Code Source de l'application SimpleTab

Romain Sauser
I.DA-P4A
CFPT INFORMATIQUE

24.05.2018

Table des matières

Chapitre 1

SimpleTab

1.1 config

1.1.1 conparam.php

```php
<?php
/**
 * Auteur: romain.ssr@eduge.ch
 * Date : 08.05.2018
 * projet: SimpleTab
 * description: Variables pour la connection à  la base de données
 */
define ('EDB_DBTYPE', "mysql");
define ('EDB_HOST', "localhost");
define ('EDB_PORT', "3306");
define ('EDB_DBNAME', "simpletab");
define ('EDB_USER', "root");
define ('EDB_PASS', "root");

?>
```

1.2 controller

1.2.1 getUserByEmail.php

```php
<?php
/**
 * Auteur: romain.ssr@eduge.ch
 * Date : 24.05.2018
 * projet: SimpleTab
 * description : récupère un utilisateur depuis son email
 */

require_once '../model/userManager.php';

// Nécessaire lorsqu'on retourne du json
header('Content-Type: application/json');

// Je récupère l'id de l'utilisateur
$email = "";


if (isset($_POST['email']))
{
    $email = $_POST['email'];
}

if ($email != ""){
    $user = userManager::getInstance()->getUserByEmail($email);
    if ($user === false){
        return false; // le pseudo est libre
        exit();
    }

    $jsn = json_encode($user);
    // Problème d'encodage Json
    if ($jsn == FALSE){
        $code = json_last_error();
        echo '{ "ReturnCode": 3, "Message": "Un problème de d\'encodage json ('.$code.'"}';
        exit();
    }
    // Succès
```

```
    echo '{ "ReturnCode": 0, "Data": '.utf8_encode($jsn).'}';
    exit();

}

// Erreur
echo '{ "ReturnCode": 1, "Message": "Il manque les paramètres"}';

?>
```

1.2.2 getUserByPseudo.php

```
<?php
/**
 * Auteur: romain.ssr@eduge.ch
 * Date : 24.05.2018
 * projet: SimpleTab
 * description : Récupère l'utilisateur par son pseudo
 */

require_once '../model/userManager.php';

// Nécessaire lorsqu'on retourne du json
header('Content-Type: application/json');

// Je récupère l'id de l'utilisateur
$pseudoUser = "";


if (isset($_POST['pseudo']))
{
    $pseudoUser = $_POST['pseudo'];
}

if ($pseudoUser != ""){
    $user = userManager::getInstance()->getUserByPseudo($pseudoUser);
    if ($user === false){
        return false; // le pseudo est libre
        exit();
    }

    $jsn = json_encode($user);
    // Problème d'encodage Json
    if ($jsn == FALSE){
        $code = json_last_error();
        echo '{ "ReturnCode": 3, "Message": "Un problème de d\'encodage json ('.$code.'"}';
        exit();
    }
    // Succès
    echo '{ "ReturnCode": 0, "Data": '.utf8_encode($jsn).'}';
    exit();

}

// Erreur
echo '{ "ReturnCode": 1, "Message": "Il manque les paramètres"}';

?>
```

1.2.3 getRateByTabId.php

```
<?php
/**
 * Auteur: romain.ssr@eduge.ch
 * Date : 23.05.2018
 * projet: SimpleTab
 */

/**
 * @copyright romain.ssr@eduge.ch 2018
 * @brief Récupère les informations des notes par l'id d'une tablature
 */




require_once "../model/rateManager.php";

// Je récupère l'id de la tablature
$idTab = -1;


if (isset($_POST['idTab']))
{
    $idTab = $_POST['idTab'];
}
```

```php

if ($idTab != -1) {

    $rate = rateManager::getInstance()->getRateByTabId($idTab);
}
if ($rate === false){
    echo '{ "ReturnCode": 2, "Message": "Un problème de récupération des données"}';
    exit();
}

$jsn = json_encode($rate);
// Problème d'encodage Json
if ($jsn == FALSE){
    $code = json_last_error();
    echo '{ "ReturnCode": 3, "Message": "Un problème de d\'encodage json ('.$code.'"}';
    exit();
}
// Succès
echo '{ "ReturnCode": 0, "Data": '.utf8_encode($jsn).'}';
exit();


?>
```

1.2.4 deleteUserById.php

```php
<?php
/**
 * Auteur: romain.ssr@eduge.ch
 * Date : 22.05.2018
 * projet: SimpleTab
 */


/**
 * @copyright romain.ssr@eduge.ch 2018
 * @brief Supprime l'utilisateur à  partir d'un ID
 */

require_once '../model/userManager.php';

// Nécessaire lorsqu'on retourne du json
header('Content-Type: application/json');

// Je récupère l'id de l'utilisateur
$idUser = -1;


if (isset($_POST['idUser']))
{
    $idUser = $_POST['idUser'];

}

if ($idUser != -1){
    $success = userManager::getInstance()->deleteUserById($idUser);
    if ($success === false){
        echo '{ "ReturnCode": 2, "Message": "Un problème de récupération des données"}';
        exit();
    }

    $jsn = json_encode($success);
    // Problème d'encodage Json
    if ($jsn == FALSE){
        $code = json_last_error();
        echo '{ "ReturnCode": 3, "Message": "Un problème de d\'encodage json ('.$code.'"}';
        exit();
    }
    // Succès
    echo '{ "ReturnCode": 0, "Data": '.utf8_encode($jsn).'}';
    exit();

}

// Erreur
echo '{ "ReturnCode": 1, "Message": "Il manque les paramètres"}';

?>
```

1.2.5 getInfosTab.php

```php
<?php
/**
 * Auteur: romain.ssr@eduge.ch
 * Date : 22.05.2018
 * projet: SimpleTab
```

```
 */

/**
 * @copyright romain.ssr@eduge.ch 2018
 * @brief Récupère les informations des tablatures au format XML
 */

session_start();


require_once "../model/tablatureManager.php";

// Je récupère l'id de la tablature
$idTab = -1;


if (isset($_POST['idTab']))
{
    $idTab = $_POST['idTab'];
    $_SESSION['currentIdTab'] = $idTab;
}

if ($idTab != -1){

    require_once "../tabs/" . $idTab . ".php";

    $tabs = new SimpleXMLElement($xmlstr);
    $title = $tabs->metadata->title;
    $author = $tabs->metadata->author;
    $tuning = $tabs->metadata->tuning;
    $capo = $tabs->metadata->capo;
    $key = $tabs->metadata->key;
    $lvl = $tabs->metadata->level;
    $bodyTab = $tabs->corpse;

    $jsn = json_encode($tabs);
    // Problème d'encodage Json
    if ($jsn == FALSE){
        $code = json_last_error();
        echo '{ "ReturnCode": 3, "Message": "Un problème de d\'encodage json ('.$code.')"}';
        exit();
    }
    // Succès
    echo '{ "ReturnCode": 0, "Data": '.utf8_encode($jsn).'}';
    exit();

}

?>
```

1.2.6 getTabAndRelatedArtistByName.php

```
<?php
/**
 * Auteur: romain.ssr@eduge.ch
 * Date : 17.05.2018
 * projet: SimpleTab
 */

/**
 * @copyright romain.ssr@eduge.ch 2018
 * @brief Récupère les tablatures ainsi que leur artiste posté par un utilisateur par leurs nom.
 */

require_once '../model/tablatureManager.php';

// Nécessaire lorsqu'on retourne du json
header('Content-Type: application/json');

$tabTitleOrArtistNAme="";

if(isset($_POST['tabTitleOrArtistNAme']))
{
    $tabTitleOrArtistNAme = $_POST['tabTitleOrArtistNAme'];
}

if($tabTitleOrArtistNAme!="")
{
    $tablature = TablatureManager::getInstance()->getTabAndRelatedArtistByName($tabTitleOrArtistNAme);
}

if ($tablature === false){
    echo '{ "ReturnCode": 2, "Message": "Un problème de récupération des données"}';
    exit();
}

$jsn = json_encode($tablature);
// Problème d'encodage Json
if ($jsn == FALSE){
    $code = json_last_error();
```

```
    echo '{ "ReturnCode": 3, "Message": "Un problème de d\'encodage json ('.$code.'"}';
    exit();
}
// Succès
echo '{ "ReturnCode": 0, "Data": '.utf8_encode($jsn).'}';
exit();


?>
```

1.2.7 updateTabRate.php

```
<?php
/**
 * Auteur: romain.ssr@eduge.ch
 * Date : 22.05.2018
 * projet: SimpleTab
 */

/**
 * @copyright romain.ssr@eduge.ch 2018
 * @brief Récupère les informations des tablatures au format XML
 */



require_once "../model/tablatureManager.php";

// Je récupère l'id de la tablature
$idTab = -1;


if (isset($_POST['idTab']))
{
    $idTab = $_POST['idTab'];
}

if ($idTab != -1){

    $success = tablatureManager::getInstance()->updateTabRate($idTab);

    $jsn = json_encode($tabs);
    // Problème d'encodage Json
    if ($jsn == FALSE){
        $code = json_last_error();
        echo '{ "ReturnCode": 3, "Message": "Un problème de d\'encodage json ('.$code.'"}';
        exit();
    }
    // Succès
    echo '{ "ReturnCode": 0, "Data": '.utf8_encode($jsn).'}';
    exit();

}

?>
```

1.2.8 approuveTab.php

```
<?php
/**
 * Auteur: romain.ssr@eduge.ch
 * Date : 22.05.2018
 * projet: SimpleTab
 */

/**
 * @copyright romain.ssr@eduge.ch 2018
 * @brief Met l'état approuvé d'une tablature à  true
 */

require_once '../model/tablatureManager.php';

// Nécessaire lorsqu'on retourne du json
header('Content-Type: application/json');

// Je récupère l'id de la tablature
$idTab = -1;


if (isset($_POST['idTab']))
{
    $idTab = $_POST['idTab'];

}

if ($idTab != -1){
```

```
    $success = TablatureManager::getInstance()->approuveTab($idTab);
    if ($success === false){
        echo '{ "ReturnCode": 2, "Message": "Un problème de récupération des données"}';
        exit();
    }

    $jsn = json_encode($success);
    // Problème d'encodage Json
    if ($jsn == FALSE){
        $code = json_last_error();
        echo '{ "ReturnCode": 3, "Message": "Un problème de d\'encodage json ('.$code.'"}';
        exit();
    }
    // Succès
    echo '{ "ReturnCode": 0, "Data": '.utf8_encode($jsn).'}';
    exit();

}

// Erreur
echo '{ "ReturnCode": 1, "Message": "Il manque les paramètres"}';

?>
```

1.2.9 getAllNonApprouvedTab.php

```
<?php
/**
 * Auteur: romain.ssr@eduge.ch
 * Date : 22.05.2018
 * projet: SimpleTab
 */

/**
 * @copyright romain.ssr@eduge.ch 2018
 * @brief Récupère toutes les tablatures dont le champs approuved = 0
 */


require_once '../model/tablatureManager.php';

// Nécessaire lorsqu'on retourne du json
header('Content-Type: application/json');


$tablature = TablatureManager::getInstance()->getAllNonApprouvedTab();
if ($tablature === false){
    echo '{ "ReturnCode": 2, "Message": "Un problème de récupération des données"}';
    exit();
}

$jsn = json_encode($tablature);
// Problème d'encodage Json
if ($jsn == FALSE){
    $code = json_last_error();
    echo '{ "ReturnCode": 3, "Message": "Un problème de d\'encodage json ('.$code.'"}';
    exit();
}
// Succès
echo '{ "ReturnCode": 0, "Data": '.utf8_encode($jsn).'}';
exit();


?>
```

1.2.10 getUserById.php

```
<?php
/**
 * Auteur: romain.ssr@eduge.ch
 * Date : 22.05.2018
 * projet: SimpleTab
 */

/**
 * @copyright romain.ssr@eduge.ch 2018
 * @brief Récupère l'utilisateur depuis son id
 */

require_once '../model/userManager.php';

// Nécessaire lorsqu'on retourne du json
header('Content-Type: application/json');

// Je récupère l'id de l'utilisateur
$idUser = "";
```

```php

if (isset($_POST['idUser']))
{
    $idUser = $_POST['idUser'];
}

if ($idUser != ""){
    $user = userManager::getInstance()->getUserById($idUser);
    if ($user === false){
        echo '{ "ReturnCode": 2, "Message": "Un problème de récupération des données"}';
        exit();
    }

    $jsn = json_encode($user);
    // Problème d'encodage Json
    if ($jsn == FALSE){
        $code = json_last_error();
        echo '{ "ReturnCode": 3, "Message": "Un problème de d\'encodage json ('.$code.')"}';
        exit();
    }
    // Succès
    echo '{ "ReturnCode": 0, "Data": '.utf8_encode($jsn).'}';
    exit();

}

// Erreur
echo '{ "ReturnCode": 1, "Message": "Il manque les paramètres"}';

?>
```

1.2.11 identifyUser.php

```php
<?php
/**
 * Auteur: romain.ssr@eduge.ch
 * Date : 16.05.2018
 * projet: SimpleTab
 */

/**
 * @copyright romain.ssr@eduge.ch 2018
 * @brief Identifie un utilisateur depuis son pseudo ou email et son mot de passe
 */

session_start();

require_once '../model/userManager.php';

// Nécessaire lorsqu'on retourne du json
header('Content-Type: application/json');

// Je récupère les champs dont j'ai besoin
$mailOrPseudoUser = "";
$passwordUser = "";

if (isset($_POST['mailOrPseudo']) && isset($_POST['pwdConnexion']))
{
    $mailOrPseudoUser = $_POST['mailOrPseudo'];
    $passwordUser = $_POST['pwdConnexion'];
}

if ($mailOrPseudoUser != "" || $passwordUser != ""){
    $passwordHashed = UserManager::getInstance()->getPasswordFromEmailOrPseudo($mailOrPseudoUser);
    $passwordHashed =$passwordHashed[0]['pwdUser'];
    $success = password_verify($passwordUser, $passwordHashed);
    $user = userManager::getInstance()->identifyUser($mailOrPseudoUser);
    if ($success === false){
        echo '{ "ReturnCode": 2, "Message": "Un problème de récupération des données"}';
        exit();
    }
    // l'utilisateur est identifié et sa session est crée
    $_SESSION['user'] = $user;

    $jsn = json_encode($user);
    // Problème d'encodage Json
    if ($jsn == FALSE){
        $code = json_last_error();
        echo '{ "ReturnCode": 3, "Message": "Un problème de d\'encodage json ('.$code.')"}';
        exit();
    }
    // Succès
    echo '{ "ReturnCode": 0, "Data": '.utf8_encode($jsn).'}';
    exit();

}

// Erreur
echo '{ "ReturnCode": 1, "Message": "Il manque les paramètres"}';
```

```
57
58 ?>
```

1.2.12 getTabAndRelatedArtistPostedByUser.php

```php
<?php
/**
 * Auteur: romain.ssr@eduge.ch
 * Date : 16.05.2018
 * projet: SimpleTab
 */


/**
 * @copyright romain.ssr@eduge.ch 2018
 * @brief Récupère les tablatures ainsi que leur artiste posté par un utilisateur
 */

require_once '../model/tablatureManager.php';

// Nécessaire lorsqu'on retourne du json
header('Content-Type: application/json');

$userId="";

if(isset($_POST['idUser']))
{
    $userId = $_POST['idUser'];
}

if($userId!="")
{
    $tablature = TablatureManager::getInstance()->getTabAndRelatedArtistPostedByUser($userId);
}

if ($tablature === false){
    echo '{ "ReturnCode": 2, "Message": "Un problème de récupération des données"}';
    exit();
}

$jsn = json_encode($tablature);
// Problème d'encodage Json
if ($jsn == FALSE){
    $code = json_last_error();
    echo '{ "ReturnCode": 3, "Message": "Un problème de d\'encodage json ('.$code.')"}';
    exit();
}
// Succès
echo '{ "ReturnCode": 0, "Data": '.utf8_encode($jsn).'}';
exit();


?>
```

1.2.13 getUsersAndNbTabPosted.php

```php
<?php
/**
 * Auteur: romain.ssr@eduge.ch
 * Date : 22.05.2018
 * projet: SimpleTab
 */

/**
 * @copyright romain.ssr@eduge.ch 2018
 * @brief Récupère les utilisateurs et le nombre de tablatures qu'ils ont posté
 */

require_once '../model/userManager.php';


// Nécessaire lorsqu'on retourne du json
header('Content-Type: application/json');


$user = userManager::getInstance()->getUsersAndNbTabPosted();
if ($user === false){
    echo '{ "ReturnCode": 2, "Message": "Un problème de récupération des données"}';
    exit();
}

$jsn = json_encode($user);
// Problème d'encodage Json
if ($jsn == FALSE){
    $code = json_last_error();
    echo '{ "ReturnCode": 3, "Message": "Un problème de d\'encodage json ('.$code.')"}';
```

```
    exit();
}
// Succès
echo '{ "ReturnCode": 0, "Data": '.utf8_encode($jsn).'}';
exit();


?>
```

1.2.14 getRateByUserId.php

```
<?php
/**
 * Auteur: romain.ssr@eduge.ch
 * Date : 23.05.2018
 * projet: SimpleTab
 */

/**
 * @copyright romain.ssr@eduge.ch 2018
 * @brief Récupère les informations des notes par l'id d'un utilisateur
 * */




require_once "../model/rateManager.php";

// Je récupère l'id de l'utilisateur
$idUser = -1;


if (isset($_POST['idUser']))
{
    $idUser = $_POST['idUser'];
}

if ($idUser != -1) {

    $rate = rateManager::getInstance()->getRateByUserId($idUser);
}
if ($rate === false){
    echo '{ "ReturnCode": 2, "Message": "Un problème de récupération des données"}';
    exit();
}

$jsn = json_encode($rate);
// Problème d'encodage Json
if ($jsn == FALSE){
    $code = json_last_error();
    echo '{ "ReturnCode": 3, "Message": "Un problème de d\'encodage json ('.$code.'"}';
    exit();
}
// Succès
echo '{ "ReturnCode": 0, "Data": '.utf8_encode($jsn).'}';
exit();


?>
```

1.2.15 getArtistNames.php

```
<?php
/**
 * Auteur: romain.ssr@eduge.ch
 * Date : 22.05.2018
 * projet: SimpleTab
 */

/**
 * @copyright romain.ssr@eduge.ch 2018
 * @brief Récupère tous les noms des artistes
 */


require_once '../model/artistManager.php';

// Nécessaire lorsqu'on retourne du json
header('Content-Type: application/json');


$artist = artistManager::getInstance()->getArtist();
if ($artist === false){
    echo '{ "ReturnCode": 2, "Message": "Un problème de récupération des données"}';
    exit();
}
```

```php

$jsn = json_encode($artist);
// Problème d'encodage Json
if ($jsn == FALSE){
    $code = json_last_error();
    echo '{ "ReturnCode": 3, "Message": "Un problème de d\'encodage json ('.$code.'"}';
    exit();
}
// Succès
echo '{ "ReturnCode": 0, "Data": '.utf8_encode($jsn).'}';
exit();


?>
```

1.2.16 getTabById.php

```php
<?php
/**
 * Auteur: romain.ssr@eduge.ch
 * Date : 23.05.2018
 * projet: SimpleTab
 */

/**
 * @copyright romain.ssr@eduge.ch 2018
 * @brief Récupère une tablatures par son id
 */

require_once '../model/tablatureManager.php';

// Nécessaire lorsqu'on retourne du json
header('Content-Type: application/json');

// Je récupère le nom de l'artiste
$idTab = "";


if (isset($_POST['idTab']))
{
    $idTab = $_POST['idTab'];

}

if ($idTab != ""){
    $tablature = TablatureManager::getInstance()->getTabById($idTab);
    if ($tablature === false){
        echo '{ "ReturnCode": 2, "Message": "Un problème de récupération des données"}';
        exit();
    }

    $jsn = json_encode($tablature);
    // Problème d'encodage Json
    if ($jsn == FALSE){
        $code = json_last_error();
        echo '{ "ReturnCode": 3, "Message": "Un problème de d\'encodage json ('.$code.'"}';
        exit();
    }
    // Succès
    echo '{ "ReturnCode": 0, "Data": '.utf8_encode($jsn).'}';
    exit();

}

// Erreur
echo '{ "ReturnCode": 1, "Message": "Il manque les paramètres"}';

?>
```

1.2.17 modifyTabById.php

```php
<?php
/**
 * Auteur: romain.ssr@eduge.ch
 * Date : 22.05.2018
 * projet: SimpleTab
 */
/**
 * @copyright romain.ssr@eduge.ch 2018
 * @brief Modifie la tablatures à  partir d'un ID
 */

session_start();

require_once '../model/tablatureManager.php';
require_once '../model/artistManager.php';
```

```php


// Nécessaire lorsqu'on retourne du json
header('Content-Type: application/json');

// Je récupère l'id de la tablature
$idTab = -1;
$title ="";
$nameArtist="";
$lvl="";
$capo = "";
$key ="";
$tuning = "";
$tabBody="";

if (isset($_POST['modifyTitle']) && isset($_POST['modifyAuthor']) && isset($_POST['modifyLvl']) && isset($_POST['modifyCapo']) && isset($_POST['modifyKey']) && isset($_POST['modifyTuning']) && isset($_POST['modifyTabBody'])&&  isset($_SESSION['currentIdTab']))
{
    $title =$_POST['modifyTitle'];
    $nameArtist=$_POST['modifyAuthor'];
    $lvl=$_POST['modifyLvl'];
    $capo = $_POST['modifyCapo'];
    $key =$_POST['modifyKey'];
    $tuning = $_POST['modifyTuning'];
    $tabBody=$_POST['modifyTabBody'];
    $idTab = $_SESSION['currentIdTab'];

}

if ($title != "" || $nameArtist != "" || $lvl != "" || $tuning != "" || $tabBody != "" || $idTab!=-1 || $capo != "" || $key!=""|| $tabBody !="" || $idTab!=-1 ||$rateValue !=-1) {


    // Récupère le nom de l'artiste dans la base sinon le crée en base et le récupère.
    $artist = artistManager::getInstance()->getArtistByName($nameArtist);
    if ($artist == false) {
        if (!$artistAdded = artistManager::getInstance()->addArtist($nameArtist)) {
            exit();
        } else {
            $artist = artistManager::getInstance()->getArtistByName($nameArtist);
        }
    }

    $idArtist = $artist['0']['idArtist'];
    $artistName = $artist['0']['nameArtist'];
    $success = TablatureManager::getInstance()->modifyTab($idTab,$title,$artistName,$idArtist,$tuning,$capo,$key,$lvl,$tabBody);
    if ($success === false){
        echo '{ "ReturnCode": 2, "Message": "Un problème de récupération des données"}';
        exit();
    }

    $jsn = json_encode($success);
    // Problème d'encodage Json
    if ($jsn == FALSE){
        $code = json_last_error();
        echo '{ "ReturnCode": 3, "Message": "Un problème de d\'encodage json ('.$code.'"}';
        exit();
    }
    // Succès
    echo '{ "ReturnCode": 0, "Data": '.utf8_encode($jsn).'}';
    exit();

}

// Erreur
echo '{ "ReturnCode": 1, "Message": "Il manque les paramètres"}';

?>
```

1.2.18 getTabAndRelatedArtist.php

```php
<?php
/**
 * Auteur: romain.ssr@eduge.ch
 * Date : 15.05.2018
 * projet: SimpleTab
 */


/**
 * @copyright romain.ssr@eduge.ch 2018
 * @brief Récupère les tablatures et les artistes
 */

require_once '../model/tablatureManager.php';

// Nécessaire lorsqu'on retourne du json
header('Content-Type: application/json');
```

```
18
19
20          $tablature = TablatureManager::getInstance()->getTabAndRelatedArtist();
21          if ($tablature === false){
22              echo '{ "ReturnCode": 2, "Message": "Un problème de récupération des données"}';
23              exit();
24          }
25
26          $jsn = json_encode($tablature);
27          // Problème d'encodage Json
28          if ($jsn == FALSE){
29              $code = json_last_error();
30              echo '{ "ReturnCode": 3, "Message": "Un problème de d\'encodage json ('.$code.'"}';
31              exit();
32          }
33          // Succès
34          echo '{ "ReturnCode": 0, "Data": '.utf8_encode($jsn).'}';
35          exit();
36
37
38  ?>
```

1.2.19 getTabByArtist.php

```
1   <?php
2   /**
3    * Auteur: romain.ssr@eduge.ch
4    * Date : 15.05.2018
5    * projet: SimpleTab
6    */
7
8   /**
9    * @copyright romain.ssr@eduge.ch 2018
10   * @brief Récupère les tablatures d'un artiste en particulier
11   */
12
13  require_once '../model/tablatureManager.php';
14
15  // Nécessaire lorsqu'on retourne du json
16  header('Content-Type: application/json');
17
18  // Je récupère le nom de l'artiste
19  $artistName = "";
20
21
22  if (isset($_POST['artistName']))
23  {
24      $artistName = $_POST['artistName'];
25
26  }
27
28    if ($artistName != ""){
29          $tablature = TablatureManager::getInstance()->sortTabByArtist($artistName);
30          if ($tablature === false){
31              echo '{ "ReturnCode": 2, "Message": "Un problème de récupération des données"}';
32              exit();
33          }
34
35          $jsn = json_encode($tablature);
36          // Problème d'encodage Json
37          if ($jsn == FALSE){
38              $code = json_last_error();
39              echo '{ "ReturnCode": 3, "Message": "Un problème de d\'encodage json ('.$code.'"}';
40              exit();
41          }
42          // Succès
43          echo '{ "ReturnCode": 0, "Data": '.utf8_encode($jsn).'}';
44          exit();
45
46      }
47
48    // Erreur
49    echo '{ "ReturnCode": 1, "Message": "Il manque les paramètres"}';
50
51  ?>
```

1.2.20 addTab.php

```
1   <?php
2   /**
3    * Auteur: romain.ssr@eduge.ch
4    * Date : 18.05.2018
5    * projet: SimpleTab
6    */
7
8   /**
```

```php
 * @copyright romain.ssr@eduge.ch 2018
 * @brief Ajoute une tablature en base
 */

session_start();

require_once '../model/tablatureManager.php';
require_once '../model/artistManager.php';

// Nécessaire lorsqu'on retourne du json
header('Content-Type: application/json');

// Je récupère les champs dont j'ai besoin
$title ="";
$nameArtist="";
$lvl="";
$capo = "";
$key ="";
$tuning = "";
$tabBody="";


if (isset($_POST['addTitle']) && isset($_POST['addArtist']) && isset($_POST['addLvl']) && isset($_POST['addTuning']) && isset($_POST['addTabBody']) && isset($_SESSION['user'] ))
{
    $title =$_POST['addTitle'];
    $nameArtist=$_POST['addArtist'];
    $lvl=$_POST['addLvl'];
    $tuning = $_POST['addTuning'];
    $tabBody=$_POST['addTabBody'];
    $userId =$_SESSION['user'][0]['idUsers'];
}

if(!isset($_POST['addCapo']))
{
    $capo = "";
}


if(!isset($_POST['addKey']))
{
    $key = "";
}


if ($title != "" || $nameArtist != "" || $lvl != "" || $tuning != "" || $tabBody != "" ){

    // Récupère le nom de l'artiste dans la base sinon le crée en base et le récupère.
    $artist = artistManager::getInstance()->getArtistByName($nameArtist);
    if($artist == false)
    {
        if(!$artistAdded = artistManager::getInstance()->addArtist($nameArtist))
        {
            exit();
        }
        else
        {
            $artist = artistManager::getInstance()->getArtistByName($nameArtist);
        }
    }

    $idArtist = $artist['0']['idArtist'];
    $artistName = $artist['0']['nameArtist'];
    $success = tablatureManager::getInstance()->addTab($title, $lvl, $idArtist, $userId, $artistName, $capo, $key, $tuning, $tabBody);
    if ($success === false) {
        echo '{ "ReturnCode": 2, "Message": "Un problème de récupération des données"}';
        exit();
    }


    $jsn = json_encode($success);
    // Problème d'encodage Json
    if ($jsn == FALSE){
        $code = json_last_error();
        echo '{ "ReturnCode": 3, "Message": "Un problème de d\'encodage json ('.$code.'"}';
        exit();
    }
    // Succès
    echo '{ "ReturnCode": 0, "Data": '.utf8_encode($jsn).'}';
    exit();

}

// Erreur
echo '{ "ReturnCode": 1, "Message": "Il manque les paramètres"}';

?>
```

1.2.21 deleteTabById.php

```php
<?php
/**
 * Auteur: romain.ssr@eduge.ch
 * Date : 22.05.2018
 * projet: SimpleTab
 */


/**
 * @copyright romain.ssr@eduge.ch 2018
 * @brief Supprime la tablatures à  partir d'un ID
 */

require_once '../model/tablatureManager.php';

// Nécessaire lorsqu'on retourne du json
header('Content-Type: application/json');

// Je récupère l'id de la tablature
$idTab = -1;


if (isset($_POST['idTab']))
{
    $idTab = $_POST['idTab'];

}

if ($idTab != -1){
    $success = TablatureManager::getInstance()->deleteTab($idTab);
    if ($success === false){
        echo '{ "ReturnCode": 2, "Message": "Un problème de récupération des données"}';
        exit();
    }

    $jsn = json_encode($success);
    // Problème d'encodage Json
    if ($jsn == FALSE){
        $code = json_last_error();
        echo '{ "ReturnCode": 3, "Message": "Un problème de d\'encodage json ('.$code.'"}';
        exit();
    }
    // Succès
    echo '{ "ReturnCode": 0, "Data": '.utf8_encode($jsn).'}';
    exit();

}

// Erreur
echo '{ "ReturnCode": 1, "Message": "Il manque les paramètres"}';

?>
```

1.2.22 getTabByTitle.php

```php
<?php
/**
 * Auteur: romain.ssr@eduge.ch
 * Date : 17.05.2018
 * projet: SimpleTab
 */
/**
 * @copyright romain.ssr@eduge.ch 2018
 * @brief Récupère les tablatures à  partir du titre
 */

require_once '../model/tablatureManager.php';

// Nécessaire lorsqu'on retourne du json
header('Content-Type: application/json');

// Je récupère le nom de l'artiste
$titleTab = "";


if (isset($_POST['titleTab']))
{
    $titleTab = $_POST['titleTab'];

}

  if ($titleTab != ""){
        $tablature = TablatureManager::getInstance()->getTabByTitle($titleTab);
        if ($tablature === false){
            echo '{ "ReturnCode": 2, "Message": "Un problème de récupération des données"}';
            exit();
        }

        $jsn = json_encode($tablature);
```

```
        // Problème d'encodage Json
        if ($jsn == FALSE){
            $code = json_last_error();
            echo '{ "ReturnCode": 3, "Message": "Un problème de d\'encodage json ('.$code.'"}';
            exit();
        }
        // Succès
        echo '{ "ReturnCode": 0, "Data": '.utf8_encode($jsn).'}';
        exit();

    }

  // Erreur
  echo '{ "ReturnCode": 1, "Message": "Il manque les paramètres"}';

?>
```

1.2.23 addComment.php

```
<?php
/**
 * Auteur: romain.ssr@eduge.ch
 * Date : 18.05.2018
 * projet: SimpleTab
 */

/**
 * @copyright romain.ssr@eduge.ch 2018
 * @brief Ajoute un commentaire en base
 */

require_once '../model/commentManager.php';

// Nécessaire lorsqu'on retourne du json
header('Content-Type: application/json');

// Je récupère les champs dont j'ai besoin
$contentComment = "";
$idTabComment = -1;
$idUserComment = -1;




if (isset($_POST['contentComment']) && isset($_POST['idTabComment']) && isset($_POST['idUserComment']) )
{
    $contentComment = $_POST['contentComment'];
    $idTabComment = $_POST['idTabComment'];
    $idUserComment = $_POST['idUserComment'];
}

if ($contentComment != "" || $idTabComment != -1 || $idUserComment != -1 ){
    $success = commentManager::getInstance()->addComment($contentComment,$idTabComment,$idUserComment);
    if ($success === false){
        echo '{ "ReturnCode": 2, "Message": "Un problème de récupération des données"}';
        exit();
    }

    $jsn = json_encode($success);
    // Problème d'encodage Json
    if ($jsn == FALSE){
        $code = json_last_error();
        echo '{ "ReturnCode": 3, "Message": "Un problème de d\'encodage json ('.$code.'"}';
        exit();
    }
    // Succès
    echo '{ "ReturnCode": 0, "Data": '.utf8_encode($jsn).'}';
    exit();

}

// Erreur
echo '{ "ReturnCode": 1, "Message": "Il manque les paramètres"}';

?>
```

1.2.24 getDifficultyInLetters.php

```
<?php
/**
 * Auteur: romain.ssr@eduge.ch
 * Date : 15.05.2018
 * projet: SimpleTab
 */

/**
```

```
 * @copyright romain.ssr@eduge.ch 2018
 * @brief Renvoie la difficulté d'une tablature en toute lettres depuis un chiffre
 */

require_once '../model/tablatureManager.php';

// Nécessaire lorsqu'on retourne du json
header('Content-Type: application/json');

// Je récupère le nom de l'artiste
$lvlTab = "";


if (isset($_POST['lvlTab']))
{
    $lvlTab = $_POST['lvlTab'];

}

if ($lvlTab != ""){
    $difficulty = TablatureManager::getInstance()->getDifficultyInLetters($lvlTab);
    if ($difficulty === false){
        echo '{ "ReturnCode": 2, "Message": "Un problème de récupération des données"}';
        exit();
    }

    $jsn = json_encode($difficulty);
    // Problème d'encodage Json
    if ($jsn == FALSE){
        $code = json_last_error();
        echo '{ "ReturnCode": 3, "Message": "Un problème de d\'encodage json ('.$code.')"}';
        exit();
    }
    // Succès
    echo '{ "ReturnCode": 0, "Data": '.utf8_encode($jsn).'}';
    exit();

}

// Erreur
echo '{ "ReturnCode": 1, "Message": "Il manque les paramètres"}';

?>
```

1.2.25 addUser.php

```
<?php
/**
 * Auteur: romain.ssr@eduge.ch
 * Date : 15.05.2018
 * projet: SimpleTab
 */

/**
 * @copyright romain.ssr@eduge.ch 2018
 * @brief Ajoute un utilisateur en base
 */

require_once '../model/userManager.php';

// Nécessaire lorsqu'on retourne du json
header('Content-Type: application/json');

// Je récupère les champs dont j'ai besoin
$nameUser = "";
$forenameUser = "";
$passwordUser = "";
$emailUser = "";
$pseudoUser = "";



if (isset($_POST['name']) && isset($_POST['forename']) && isset($_POST['pseudo']) &&isset($_POST['email']) ↩
     &&isset($_POST['password'])&&isset($_POST['passwordConfirm']) && $_POST['password'] == ↩
     $_POST['passwordConfirm'])
{
    if (filter_var($_POST['email'], FILTER_VALIDATE_EMAIL)) {
        if (checkdnsrr(explode('@',$_POST['email'])[1], 'MX')) {
            $nameUser = $_POST['name'];
            $forenameUser = $_POST['forename'];
            $passwordUser = password_hash($_POST['password'], PASSWORD_BCRYPT);
            $emailUser = $_POST['email'];
            $pseudoUser = $_POST['pseudo'];

        }

    }

}

```

```php
if ($nameUser != "" || $forenameUser != "" || $passwordUser != "" || $emailUser != "" || $pseudoUser != "" ){
    $success = ↩
     UserManager::getInstance()->addUser($nameUser,$forenameUser,$passwordUser,$emailUser,$pseudoUser);
    if ($success === false){
        echo '{ "ReturnCode": 2, "Message": "Un problème de récupération des données"}';
        exit();
    }

    $jsn = json_encode($success);
    // Problème d'encodage Json
    if ($jsn == FALSE){
        $code = json_last_error();
        echo '{ "ReturnCode": 3, "Message": "Un problème de d\'encodage json ('.$code.'"}';
        exit();
    }
    // Succès
    echo '{ "ReturnCode": 0, "Data": '.utf8_encode($jsn).'}';
    exit();

}

// Erreur
echo '{ "ReturnCode": 1, "Message": "Il manque les paramètres"}';

?>
```

1.2.26 getCommentsByTab.php

```php
<?php
/**
 * Auteur: romain.ssr@eduge.ch
 * Date : 17.05.2018
 * projet: SimpleTab
 */


/**
 * @copyright romain.ssr@eduge.ch 2018
 * @brief Récupère les commentaires à  partir d'une tablature
 */

require_once '../model/commentManager.php';

// Nécessaire lorsqu'on retourne du json
header('Content-Type: application/json');

// Je récupère l'id de la tablature
$idTab = "";


if (isset($_POST['idTab']))
{
    $idTab = $_POST['idTab'];

}

  if ($idTab != ""){
        $comment = commentManager::getInstance()->getCommentAndUsersByTab($idTab);
        if ($comment === false){
            echo '{ "ReturnCode": 2, "Message": "Un problème de récupération des données"}';
            exit();
        }

        $jsn = json_encode($comment);
        // Problème d'encodage Json
        if ($jsn == FALSE){
            $code = json_last_error();
            echo '{ "ReturnCode": 3, "Message": "Un problème de d\'encodage json ('.$code.'"}';
            exit();
        }
        // Succès
        echo '{ "ReturnCode": 0, "Data": '.utf8_encode($jsn).'}';
        exit();

    }

  // Erreur
  echo '{ "ReturnCode": 1, "Message": "Il manque les paramètres"}';

?>
```

1.2.27 addRate.php

```php
<?php

/**
```

```
 * Auteur: romain.ssr@eduge.ch
 * Date : 22.05.2018
 * projet: SimpleTab
 */

/**
 * @copyright romain.ssr@eduge.ch 2018
 * @brief Ajoute une note dans la base
 */

require_once '../model/rateManager.php';
require_once '../model/tablatureManager.php';


// Nécessaire lorsqu'on retourne du json
header('Content-Type: application/json');

$idUser = -1;
$idTab = -1;
$rateValue = -1;

if(isset($_POST['idUser']) && isset($_POST['idTab']) && isset($_POST['rate']))
{
    $idUser = $_POST['idUser'];
    $idTab = $_POST['idTab'];
    $rateValue = $_POST['rate'];
}

if($idUser !=-1 && $idTab != -1 && $rateValue !=-1)
{
    $rate = rateManager::getInstance()->addRate($rateValue,$idTab,$idUser);
    $avergageRate = rateManager::getInstance()->getAverageRateByTabId($idTab);
    $avergageRate =$avergageRate[0][0];
    $updateRateSuccess = tablatureManager::getInstance()->updateTabRate($avergageRate,$idTab);
}

if ($rate === false && $updateRateSuccess === false){
    echo '{ "ReturnCode": 2, "Message": "Un problème de récupération des données"}';
    exit();
}

$jsn = json_encode($rate);
// Problème d'encodage Json
if ($jsn == FALSE){
    $code = json_last_error();
    echo '{ "ReturnCode": 3, "Message": "Un problème de d\'encodage json ('.$code.'"}';
    exit();
}
// Succès
echo '{ "ReturnCode": 0, "Data": '.utf8_encode($jsn).'}';
exit();


?>
```

1.2.28 destroySession.php

```
<?php
/**
 * Auteur: romain.ssr@eduge.ch
 * Date : 16.05.2018
 * projet: SimpleTab
 */

//Permet de déconnecter l'utilisateur
session_start();
session_unset();
session_destroy();
header("Location: ../view/homePage.php")
?>
```

1.3 model

1.3.1 userManager.php

```
<?php
/**
 * Auteur: romain.ssr@eduge.ch
 * Date : 15.05.2018
 * projet: SimpleTab
 */

require_once '../model/database.php';
```

```php
require_once '../model/user.php';
require_once '../model/tablatureManager.php';

/**
 * @brief Helper class pour gérer les utilisateurs du site
 *
 * @author romain.ssr@eduge.ch
 */
class UserManager
{
    private static $objInstance;

    /**
     * @brief Constructeur de la Classe - Crée un nouveau UserManager si aucun n'existe déjà .
     * Le met en privé de telle sorte à  ce que personne ne peut céer un nouveau manager depuis ' = new
     * UserManager();'
     */
    private function __construct ()
    {
        $this->users = array();
    }

    /**
     * @brief Like the constructor, we make clone private so nobody can clone
     * the instance
     */
    private function __clone ()
    {}

    /**
     * @brief Retourne notre instance ou la crée
     *
     * @return $objInstance;
     */
    public static function getInstance ()
    {
        if (!self::$objInstance)
        {
            try
            {
                self::$objInstance = new UserManager();

            } catch (Exception $e)
            {
                echo "EDataManager Error: " . $e;
            }
        }
        return self::$objInstance;
    }

    /**
     * Retourne le tableau de tous les utilisateurs
     *
     */
    public function getUsers()
    {
        return $this->users;
    }

    /**
     * @param $pseudo
     * @return utilisateur sinon false
     */
    function getUserByPseudo($pseudo)
    {
        $db = Database::getInstance();

        try {
            $sql = $db->prepare("SELECT * FROM simpletab.users WHERE users.pseudoUser = :pseudo;");
            $sql-> bindParam(':pseudo',$pseudo,PDO::PARAM_STR);
            $sql->execute();
            $result = $sql->fetchAll();
            return $result;
        }

        catch (PDOException $e) {
            return false;
        }
    }

    /**
     * @param $email
     * @return utilisateur sinon false
     */
    function getUserByEmail($email)
    {
        $db = Database::getInstance();

        try {
            $sql = $db->prepare("SELECT * FROM simpletab.users WHERE users.emailUser = :email;");
            $sql-> bindParam(':email',$email,PDO::PARAM_STR);
            $sql->execute();
            $result = $sql->fetchAll();
            return $result;
```

```
103         }
104
105         catch (PDOException $e) {
106             return false;
107         }
108     }
109
110     /**
111      * @param $emailOrPseudo
112      * @return le mot de passe de l'utilisateur hashé si succès sinon false
113      */
114     function getPasswordFromEmailOrPseudo($mailOrPseudo)
115     {
116         $db = Database::getInstance();
117
118         try {
119             $sql = $db->prepare("SELECT users.pwdUser FROM simpletab.users WHERE (users.emailUser = ↩
     :mailOrPseudo OR users.pseudoUser = :mailOrPseudo);");
120             $sql-> bindParam(':mailOrPseudo',$mailOrPseudo,PDO::PARAM_STR);
121             $sql->execute();
122             $result = $sql->fetchAll();
123             return $result;
124         }
125
126         catch (PDOException $e) {
127             return false;
128         }
129     }
130
131     /**
132      * Récupère les utilisateurs et le nombre de tablatures qu'ils ont postés
133      * @return les utilisateurs et le nombre de tablatures qu'ils ont postés, sinon false
134      */
135     function getUsersAndNbTabPosted()
136     {
137         $db = Database::getInstance();
138
139         try {
140             $sql = $db->prepare("SELECT users.idUsers, pseudoUser, emailUser, count(users_idUsers) as nbTab
141                                         FROM users LEFT JOIN tablatures ON (tablatures.users_idUsers = ↩
     users.idUsers)
142                                         WHERE users.role_idrole = 0
143                                         GROUP BY users.idUsers, pseudoUser, emailUser
144                                         ORDER BY count(users_idUsers) DESC, pseudoUser;");
145             $sql->execute();
146             $result = $sql->fetchAll();
147             return $result;
148         }
149
150         catch (PDOException $e) {
151             return false;
152         }
153     }
154
155     /**
156      * Ajoute un utilisateur en base
157      * @param $name
158      * @param $forename
159      * @param $password
160      * @param $email
161      * @param $pseudo
162      * @return true si succès sinon false
163      */
164     public function addUser($name, $forename, $password,$email,$pseudo)
165     {
166         $db = Database::getInstance();
167         $defaultRole = 0;
168         try {
169             $sql = $db->prepare("INSERT INTO users ( nameUser, forenameUser, pwdUser, emailUser, ↩
     pseudoUser,role_idrole) VALUES (  :name, :forname,:password, :email, :pseudo,:role);");
170             $sql->bindParam(':name', $name, PDO::PARAM_STR);
171             $sql->bindParam(':forname', $forename, PDO::PARAM_STR);
172             $sql->bindParam(':password', $password, PDO::PARAM_STR);
173             $sql->bindParam(':email', $email, PDO::PARAM_STR);
174             $sql->bindParam(':pseudo', $pseudo, PDO::PARAM_STR);
175             $sql->bindParam(':role', $defaultRole, PDO::PARAM_STR);
176             $sql->execute();
177             return true;
178         }
179         catch (PDOException $e) {
180             return false;
181         }
182     }
183
184     /**
185      * Efface un utilisateur en base ainsi que ses tablatures associées
186      * @param $idUser
187      * @return true si succès sinon false
188      */
189     function deleteUserById($idUser)
190     {
191         $db = Database::getInstance();
192         try {
193
```

```
194             $sql = $db->prepare("SELECT idTab FROM simpletab.tablatures JOIN simpletab.users ON ↩
    tablatures.users_idUsers = users.idUsers where users.idUsers = :idUser;");
195             $sql->bindParam(':idUser', $idUser, PDO::PARAM_INT);
196             $sql->execute();
197             $result = $sql->fetchAll();
198             if($result!= "" && isset($result) && count($result) != 0)
199             {
200                 for ($i =0; $i<count($result);$i++)
201                 {
202                     $successDeleteTab = tablatureMAnager::getInstance()->deleteTab($result[$i]['idTab']);
203 
204                 }
205             }
206             else
207             {
208                 $successDeleteTab = true;
209             }
210 
211             if($successDeleteTab)
212             {
213                 $sql = $db->prepare("DELETE  FROM simpletab.users WHERE users.idUsers = :idUser");
214                 $sql->bindParam(':idUser', $idUser, PDO::PARAM_INT);
215                 $sql->execute();
216                 return true;
217             }
218             else
219             {
220                 return false;
221             }
222         }
223         catch (PDOException $e) {
224             return false;
225         }
226     }
227 
228     /**
229      * Retourne l'utilisateur s'il existe en base  sinon false.
230      *
231      */
232     function identifyUser($mailOrPseudo)
233     {
234         $db = Database::getInstance();
235 
236         try {
237             $sql = $db->prepare("SELECT * FROM simpletab.users WHERE (users.emailUser = :mailOrPseudo OR ↩
    users.pseudoUser = :mailOrPseudo);");
238             $sql-> bindParam(':mailOrPseudo',$mailOrPseudo,PDO::PARAM_STR);
239             $sql-> bindParam(':password',$password,PDO::PARAM_STR);
240             $sql->execute();
241             $result = $sql->fetchAll();
242             return $result;
243         }
244 
245         catch (PDOException $e) {
246             return false;
247         }
248     }
249 
250 
251     /** Tableau d'utilisateurs */
252     private $users;
253 }
254 ?>
```

1.3.2 tablatureManager.php

```
1  <?php
2  /**
3   * Auteur: romain.ssr@eduge.ch
4   * Date : 15.05.2018
5   * projet: SimpleTab
6   */
7 
8  require_once '../model/database.php';
9  require_once '../model/tablature.php';
10 require_once '../model/commentManager.php';
11 require_once '../model/rateManager.php';
12 
13 
14 
15 /**
16  * @brief Helper class pour gérer les tablatures du site
17  *
18  * @author romain.ssr@eduge.ch
19  */
20 class TablatureManager
21 {
22     private static $objInstance;
23 
24     /**
```

```
25      * @brief Class Constructor - Create a new ECommentManager if one doesn't exist
26      * Set to private so no-one can create a new instance via ' = new
27      * ECommentManager();'
28      */
29     private function __construct ()
30     {
31         $this->tablature = array();
32     }
33
34     /**
35      * @brief Like the constructor, we make clone private so nobody can clone
36      * the instance
37      */
38     private function __clone ()
39     {}
40
41     /**
42      * @brief Retourne notre instance ou la crée
43      *
44      * @return $objInstance;
45      */
46     public static function getInstance ()
47     {
48         if (!self::$objInstance)
49         {
50             try
51             {
52                 self::$objInstance = new TablatureManager();
53
54             } catch (Exception $e)
55             {
56                 echo "EDataManager Error: " . $e;
57             }
58         }
59         return self::$objInstance;
60     }
61
62     /**
63      * Retourne le tableau de toutes les tablatures
64      *
65      */
66     public function getTablature()
67     {
68         return $this->tablature;
69     }
70
71
72     function getTabById($idTab)
73     {
74         $db = Database::getInstance();
75
76         try {
77             $sql = $db->prepare("SELECT * FROM simpletab.tablatures WHERE tablatures.idTab = :idTab;");
78             $sql->bindParam("idTab", $idTab,PDO::PARAM_INT);
79             $sql->execute();
80             $result = $sql->fetchAll();
81             return $result;
82         }
83
84         catch (PDOException $e) {
85             return false;
86         }
87     }
88
89     /**
90      * Retourne Un tableau contenant les tablatures et les artistes associés ou false si une erreur est ↩
       survenue
91      *
92      */
93     function getTabAndRelatedArtist()
94     {
95         $db = Database::getInstance();
96
97         try {
98             $sql = $db->prepare("SELECT * FROM simpletab.tablatures JOIN simpletab.artists ON ↩
       tablatures.ARTISTS_idArtist = artists.idArtist WHERE tablatures.approuved = 1;");
99             $sql->execute();
100            $result = $sql->fetchAll();
101            return $result;
102        }
103
104        catch (PDOException $e) {
105            return false;
106        }
107    }
108
109    /**
110     * Retourne Un tableau contenant les tablatures et les artistes associés que l'utilisateur a posté ou ↩
       false si une erreur est survenue
111     * @param $userId -> l'identifiant de l'utilisateur
112     */
113    function getTabAndRelatedArtistPostedByUser($userId)
114    {
115        $db = Database::getInstance();
```

```
116
117         try {
118             $sql = $db->prepare("SELECT * FROM simpletab.tablatures JOIN simpletab.artists ON ↩
        tablatures.ARTISTS_idArtist = artists.idArtist WHERE (tablatures.users_idUsers = :userId AND ↩
        tablatures.approuved = 1);");
119             $sql->bindParam(':userId',$userId,PDO::PARAM_INT);
120             $sql->execute();
121             $result = $sql->fetchAll();
122             return $result;
123         }
124
125         catch (PDOException $e) {
126             return false;
127         }
128     }
129
130     /**
131      * Récupère toutes les tablatures non approuvée (champ approuved =0)
132      * @return un tableau des tablatures non approuvées ou false si échoue
133      */
134     function getAllNonApprouvedTab()
135     {
136         $db = Database::getInstance();
137
138         try {
139             $sql = $db->prepare("SELECT * FROM simpletab.tablatures JOIN simpletab.artists ON ↩
        tablatures.ARTISTS_idArtist = artists.idArtist WHERE tablatures.approuved = 0;");
140             $sql->execute();
141             $result = $sql->fetchAll();
142             return $result;
143         }
144
145         catch (PDOException $e) {
146             return false;
147         }
148     }
149
150     /**
151      * Retourne les tablatures associées à  un artiste, sinon false
152      *
153      * @param $artistName
154      *
155      */
156     function sortTabByArtist($artistName)
157     {
158         $db = Database::getInstance();
159
160         try {
161             $sql = $db->prepare("SELECT * FROM simpletab.tablatures JOIN simpletab.artists ON ↩
        tablatures.ARTISTS_idArtist = artists.idArtist WHERE  artists.nameArtist = :nameArtist;");
162             $sql->bindParam(':nameArtist', $artistName, PDO::PARAM_STR);
163             $sql->execute();
164             $result = $sql->fetchAll();
165             return $result;
166         }
167
168         catch (PDOException $e) {
169             return false;
170         }
171     }
172
173     /**
174      * @param $tabTitleOrArtistNAme
175      * @return la tablature qui correspond au nom de l'artiste ou au titre de la tablature, sinon false
176      */
177     function getTabAndRelatedArtistByName($tabTitleOrArtistNAme)
178     {
179         $db = Database::getInstance();
180
181         try {
182             $sql = $db->prepare("SELECT * FROM simpletab.tablatures JOIN simpletab.artists ON ↩
        tablatures.ARTISTS_idArtist = artists.idArtist
183                                             WHERE tablatures.titleTab = :tabTitleOrArtistNAme OR ↩
        artists.nameArtist = :tabTitleOrArtistNAme;");
184             $sql->bindParam(':tabTitleOrArtistNAme',$tabTitleOrArtistNAme,PDO::PARAM_STR);
185             $sql->execute();
186             $result = $sql->fetchAll();
187             return $result;
188         }
189
190         catch (PDOException $e) {
191             return false;
192         }
193     }
194
195     /**
196      * @param $titleTab
197      * @return Un tableau contant les informations de la tablature récupérée par son titre
198      */
199     function getTabByTitle($titleTab)
200     {
201         $db = Database::getInstance();
202
203         try {
```

```
204             $sql = $db->prepare("SELECT * FROM simpletab.tablatures  WHERE tablatures.titleTab = :titleTab;");
205             $sql->bindParam(':titleTab',$titleTab,PDO::PARAM_STR);
206             $sql->execute();
207             $result = $sql->fetchAll();
208             return $result;
209         }
210 
211         catch (PDOException $e) {
212             return false;
213         }
214     }
215 
216     /**
217      * Ajoute une tablature en base et dans un fichier php au format XML
218      * @param $title
219      * @param $lvl
220      * @param $artistId
221      * @param $userId
222      * @param $artistName
223      * @param $capo
224      * @param $key
225      * @param $tuning
226      * @param $tabBody
227      * @return true si la tablature a été correctement ajoutée en base et que le fichier xml de la ↩
     tablature a bien été créé, false sinon
228      */
229     function addTab($title, $lvl, $artistId, $userId, $artistName, $capo, $key, $tuning, $tabBody)
230     {
231         $db = Database::getInstance();
232         $path = "temp.php";
233         try {
234             $db->beginTransaction();
235 
236             $sql = $db->prepare("INSERT INTO simpletab.tablatures (titleTab, pathTab, lvlTab, ↩
     ARTISTS_idArtist, users_idUsers, approuved) VALUES (:title, :path, :lvl, :artistId, :idUser, 0);");
237             $sql->bindParam(':title', $title, PDO::PARAM_STR);
238             $sql->bindParam(':path', $path, PDO::PARAM_STR);
239             $sql->bindParam(':lvl', $lvl, PDO::PARAM_STR);
240             $sql->bindParam(':artistId', $artistId, PDO::PARAM_STR);
241             $sql->bindParam(':idUser', $userId, PDO::PARAM_STR);
242             $sql->execute();
243 
244             $lastId = $db->lastInsertId();
245             $newPath = $lastId.".php";
246 
247             $sql = $db->prepare("UPDATE simpletab.tablatures SET pathTab = :path WHERE idTab = :idTab;");
248             $sql->bindParam(':idTab', $lastId, PDO::PARAM_STR);
249             $sql->bindParam(':path',$newPath , PDO::PARAM_STR);
250             $sql->execute();
251 
252             $newPath = "../tabs/$lastId.php";
253             $tab = fopen($newPath,'a+');
254             $lvl = $this->getDifficultyInLetters($lvl);
255 
256             $tabXml = '<?php
257 $xmlstr = <<<XML
258 <?xml version = "1.0" encoding="UTF-8" standalone="yes" ?>
259 <tabs>
260     <metadata>
261             <title> '.$title.' </title>
262             <author>'.$artistName.'</author>
263             <tuning>'.$tuning.'</tuning>
264             <capo>'.$capo.'</capo>
265             <key>'.$key.'</key>
266             <level> '.$lvl.'</level>
267     </metadata>
268     <corpse>
269     '.$tabBody.'
270     </corpse>
271 </tabs>
272 XML;
273 ?>';
274             $tabXml = str_replace('&','&amp;',$tabXml);
275             ftruncate($tab,0);
276             fputs($tab,$tabXml);
277             fclose($tab);
278 
279             $db->commit();
280             return true;
281         }
282 
283         catch (PDOException $e) {
284             $db->rollBack();
285             return false;
286         }
287     }
288 
289     /**
290      * Modifie une tablature en base ainsi que dans le XML
291      * @param $idTab
292      * @param $title
293      * @param $artistName
294      * @param $artistId
295      * @param $tuning
```

```
296         * @param $capo
297         * @param $key
298         * @param $lvl
299         * @param $tabBody
300         * @return true si succès sinon false
301         */
302        function modifyTab($idTab, $title, $artistName,$artistId, $tuning, $capo, $key, $lvl, $tabBody)
303        {
304            $db = Database::getInstance();
305
306            try {
307                $db->beginTransaction();
308
309                $sql = $db->prepare("UPDATE simpletab.tablatures SET titleTab = :title, lvlTab = :lvl, ↩
    ARTISTS_idArtist = :idArtist, approuved = 0 WHERE (idTab = :idTab);");
310                $sql->bindParam(':title', $title, PDO::PARAM_STR);
311                $sql->bindParam(':idArtist', $artistId, PDO::PARAM_STR);
312                $sql->bindParam(':lvl', $lvl, PDO::PARAM_INT);
313                $sql->bindParam(':idTab', $idTab, PDO::PARAM_STR);
314                $sql->execute();
315
316                $path = "../tabs/$idTab.php";
317
318
319                $tab = fopen($path,'w');
320                $lvl = $this->getDifficultyInLetters($lvl);
321
322                $tabXml = '<?php
323    $xmlstr = <<<XML
324    <?xml version = "1.0" encoding="UTF-8" standalone="yes" ?>
325    <tabs>
326         <metadata>
327                <title> '.$title.' </title>
328                <author>'.$artistName.'</author>
329                <tuning>'.$tuning.'</tuning>
330                <capo>'.$capo.'</capo>
331                <key>'.$key.'</key>
332                <level> '.$lvl.'</level>
333         </metadata>
334         <corpse>
335         '.$tabBody.'
336         </corpse>
337    </tabs>
338    XML;
339    ?>';
340                $tabXml = str_replace('&','&amp;',$tabXml);
341                ftruncate($tab,0);
342                fputs($tab,$tabXml);
343                fclose($tab);
344
345                $db->commit();
346                return true;
347            }
348
349            catch (PDOException $e) {
350                $db->rollBack();
351                return false;
352            }
353        }
354
355        /**
356         * Supprime une tablature de la base ainsi que son fichier XML
357         * @param $idTab
358         * @return bool
359         */
360        function deleteTab($idTab)
361        {
362            $db = Database::getInstance();
363            $path = "../tabs/$idTab.php";
364
365            try {
366                $db->beginTransaction();
367
368                commentManager::getInstance()->deleteCommentByTabId($idTab);
369                rateManager::getInstance()->deleteRateByIdTab($idTab);
370
371
372                $sql = $db->prepare("DELETE FROM simpletab.tablatures WHERE (idTab = :idTab);");
373                $sql->bindParam(':idTab', $idTab, PDO::PARAM_INT);
374                $sql->execute();
375                $result =unlink($path);
376                if($result)
377                {
378                    $db->commit();
379                    return true;
380                }
381                else
382                {
383                    $db->rollBack();
384                }
385            }
386            catch (PDOException $e) {
387                $db->rollBack();
388                return false;
```

```
389         }
390     }
391 
392     /**
393      * Passe le champ d'une tablature approuved à  1 ce qui la rend visible pour les utilisateurs
394      * @param $idTab
395      * @return bool
396      */
397     function approuveTab($idTab)
398     {
399         $db = Database::getInstance();
400 
401         try {
402             $sql = $db->prepare("UPDATE simpletab.tablatures SET approuved = 1 WHERE (idTab = :idTab);");
403             $sql->bindParam(':idTab',$idTab,PDO::PARAM_INT);
404             $sql->execute();
405             return true;
406         }
407 
408         catch (PDOException $e) {
409             return false;
410         }
411     }
412 
413     function updateTabRate($averageRate, $idTab)
414     {
415         $db = Database::getInstance();
416 
417         try {
418             $sql = $db->prepare("UPDATE simpletab.tablatures SET rateTab = :averageRate WHERE (idTab = ↩
     :idTab);");
419             $sql->bindParam(':idTab',$idTab,PDO::PARAM_INT);
420             $sql->bindParam(':averageRate',$averageRate,PDO::PARAM_INT);
421             $sql->execute();
422             return true;
423         }
424 
425         catch (PDOException $e) {
426             return false;
427         }
428     }
429 
430     /**
431      * @param $lvl
432      * @return la difficulté en lettre à  partir d'un chiffe
433      */
434    function getDifficultyInLetters($lvl)
435     {
436         switch ($lvl)
437         {
438             case "0":
439                 return "Facile";
440                 break;
441             case "1":
442                 return "Moyen";
443                 break;
444 
445             case "2":
446                 return "Difficile";
447                 break;
448 
449         }
450     }
451 
452     /** Tableau de tablature */
453     private $tablature;
454 }
```

1.3.3 commentManager.php

```
1  <?php
2  /**
3   * Auteur: romain.ssr@eduge.ch
4   * Date : 17.05.2018
5   * projet: SimpleTab
6   */
7 
8 
9  require_once '../model/database.php';
10 require_once '../model/comment.php';
11 
12 /**
13  * @brief Helper class pour gérer les tablatures du site
14  *
15  * @author romain.ssr@eduge.ch
16  */
17 class CommentManager
18 {
19     private static $objInstance;
20 
```

```php
    /**
     * @brief Class Constructor - Create a new ECommentManager if one doesn't exist
     * Set to private so no-one can create a new instance via ' = new
     * ECommentManager();'
     */
    private function __construct ()
    {
        $this->comments = array();
    }

    /**
     * @brief Like the constructor, we make clone private so nobody can clone
     * the instance
     */
    private function __clone ()
    {}

    /**
     * @brief Retourne notre instance ou la crée
     *
     * @return $objInstance;
     */
    public static function getInstance ()
    {
        if (!self::$objInstance)
        {
            try
            {
                self::$objInstance = new CommentManager();

            } catch (Exception $e)
            {
                echo "EDataManager Error: " . $e;
            }
        }
        return self::$objInstance;
    }


    /**
     * @param $idTab -> l'id de la tablature associée au commentaire
     * @return les commentaires associés à  la tablature sous forme de tableau ou false si une erreur survient
     */
    function getCommentAndUsersByTab($idTab)
    {
        $db = Database::getInstance();

        try {
            $sql = $db->prepare("SELECT * FROM simpletab.comments JOIN simpletab.users ON comments.users_idUsers = users.idUsers WHERE comments.tablatures_idTab = :idTab;");
            $sql->bindParam(':idTab',$idTab,PDO::PARAM_INT);
            $sql->execute();
            $result = $sql->fetchAll();
            return $result;
        }

        catch (PDOException $e) {
            return false;
        }
    }

    /**
     * Ajoute un commentaire en base
     * @param $contentCommment
     * @param $tabIdComment
     * @param $userIdComment
     * @return bool
     */
    function addComment($contentCommment, $tabIdComment, $userIdComment)
    {
        $db = Database::getInstance();

        try {
            $sql = $db->prepare("INSERT INTO simpletab.comments ( contentComment, tablatures_idTab, users_idUsers) VALUES ( :contentComment, :idTab, :idUser);");
            $sql->bindParam(':contentComment', $contentCommment, PDO::PARAM_STR);
            $sql->bindParam(':idTab', $tabIdComment, PDO::PARAM_INT);
            $sql->bindParam(':idUser', $userIdComment, PDO::PARAM_INT);
            $sql->execute();
            return true;
        }
        catch (PDOException $e) {
            return false;
        }
    }

    /**
     * Efface un commentaire par son Id
     * @param $idComment
     * @return bool
     */
    function deleteCommentByTabId($idTab)
    {
        $db = Database::getInstance();
```

```
        try{
            $sql = $db->prepare("DELETE  FROM simpletab.comments WHERE comments.tablatures_idTab = :idTab");
            $sql->bindParam(':idTab', $idTab, PDO::PARAM_INT);
            $sql->execute();
            return true;
        }
        catch (PDOException $e) {
            return false;
        }

    }

    /** Tableau de commentaires */
    private $comments;
}
```

1.3.4 rateManager.php

```
<?php
/**
 * Auteur: romain.ssr@eduge.ch
 * Date : 22.05.2018
 * projet: SimpleTab
 */

require_once '../model/database.php';
require_once '../model/rate.php';

/**
 * @brief Helper class pour gérer les artistes du site
 *
 * @author romain.ssr@eduge.ch
 */
class rateManager
{
    private static $objInstance;

    /**
     * @brief Class Constructor - Create a new ECommentManager if one doesn't exist
     * Set to private so no-one can create a new instance via ' = new
     * ECommentManager();'
     */
    private function __construct()
    {
        $this->rate = array();
    }

    /**
     * @brief Like the constructor, we make clone private so nobody can clone
     * the instance
     */
    private function __clone()
    {
    }

    /**
     * @brief Retourne notre instance ou la crée
     *
     * @return $objInstance;
     */
    public static function getInstance()
    {
        if (!self::$objInstance) {
            try {
                self::$objInstance = new rateManager();

            } catch (Exception $e) {
                echo "EDataManager Error: " . $e;
            }
        }
        return self::$objInstance;
    }

    /**
     * @return le tableau des notes sinon false
     */
    function getRate()
    {

        $db = Database::getInstance();

        try {
            $sql = $db->prepare("SELECT * FROM simpletab.rates");
            $sql->execute();
            $result = $sql->fetchAll();
            return $result;
        }
        catch (PDOException $e) {
            return false;
        }
```

```
73      }
74
75      /**
76       * Ajoute une note en base
77       * @param $rate
78       * @param $idTab
79       * @param $idUser
80       * @return true si succès, sinon false
81       */
82      function addRate($rate,$idTab,$idUser)
83      {
84          $db = Database::getInstance();
85
86          try {
87              $sql = $db->prepare("INSERT INTO simpletab.rates (rate,tablatures_idTab,users_idUsers) VALUES ( ↩
     :rate, :idTab, :idUser);");
88              $sql->bindParam(':rate', $rate, PDO::PARAM_INT);
89              $sql->bindParam(':idTab', $idTab, PDO::PARAM_INT);
90              $sql->bindParam(':idUser', $idUser, PDO::PARAM_INT);
91              $sql->execute();
92              return true;
93          }
94          catch (PDOException $e) {
95              return false;
96          }
97      }
98
99      /**
100      * Récupère la moyenne des notes par l'id d'une tablature
101      * @param $idTab
102      * @return les notes de la tablature si succès sinon false
103      */
104     function getRateByTabId($idTab)
105     {
106         $db = Database::getInstance();
107         try{
108             $sql = $db->prepare("SELECT * FROM simpletab.rates WHERE rates.tablatures_idTab = :idTab");
109             $sql->bindParam(':idTab', $idTab, PDO::PARAM_INT);
110             $sql->execute();
111             $result = $sql->fetchAll();
112             return $result;
113         }
114         catch (PDOException $e) {
115             return false;
116         }
117     }
118
119     /**
120      * Récupère la moyenne des notes par l'id d'un utilisateur
121      * @param $idTab
122      * @return les notes de la tablature si succès sinon false
123      */
124     function getRateByUserId($idUser)
125     {
126         $db = Database::getInstance();
127         try{
128             $sql = $db->prepare("SELECT * FROM simpletab.rates WHERE rates.users_idUsers = :idUser");
129             $sql->bindParam(':idUser', $idUser, PDO::PARAM_INT);
130             $sql->execute();
131             $result = $sql->fetchAll();
132             return $result;
133         }
134         catch (PDOException $e) {
135             return false;
136         }
137     }
138
139     /**
140      * Supprime une note par l'id d'une tablature
141      * @param $idTab
142      * @return true si succès sinon false
143      */
144     function deleteRateByIdTab($idTab)
145     {
146         $db = Database::getInstance();
147         try{
148             $sql = $db->prepare("DELETE  FROM simpletab.rates WHERE rates.tablatures_idTab = :idTab");
149             $sql->bindParam(':idTab', $idTab, PDO::PARAM_INT);
150             $sql->execute();
151             return true;
152         }
153         catch (PDOException $e) {
154             return false;
155         }
156     }
157
158     function getAverageRateByTabId($idTab)
159     {
160         $db = Database::getInstance();
161         try{
162             $sql = $db->prepare("SELECT ROUND(AVG(rates.rate)) FROM simpletab.rates WHERE ↩
     rates.tablatures_idTab = :idTab");
163             $sql->bindParam(':idTab', $idTab, PDO::PARAM_INT);
164             $sql->execute();
```

```
            $result = $sql->fetchAll();
            return $result;
        }
        catch (PDOException $e) {
            return false;
        }
    }

    /**
     * @var Tableau des artistes
     */
    private $rate;
}
```

1.3.5 artistManager.php

```
<?php
/**
 * Auteur: romain.ssr@eduge.ch
 * Date : 18.05.2018
 * projet: SimpleTab
 */

require_once '../model/database.php';
require_once '../model/artist.php';

/**
 * @brief Helper class pour gérer les artistes du site
 *
 * @author romain.ssr@eduge.ch
 */
class artistManager
{
    private static $objInstance;

    /**
     * @brief Class Constructor - Create a new ECommentManager if one doesn't exist
     * Set to private so no-one can create a new instance via ' = new
     * ECommentManager();'
     */
    private function __construct()
    {
        $this->artist = array();
    }

    /**
     * @brief Like the constructor, we make clone private so nobody can clone
     * the instance
     */
    private function __clone()
    {
    }

    /**
     * @brief Retourne notre instance ou la crée
     *
     * @return $objInstance;
     */
    public static function getInstance()
    {
        if (!self::$objInstance) {
            try {
                self::$objInstance = new artistManager();

            } catch (Exception $e) {
                echo "EDataManager Error: " . $e;
            }
        }
        return self::$objInstance;
    }

    /**
     * @return le tableau des artistes sinon false
     */
    function getArtist()
    {

        $db = Database::getInstance();

        try {
            $sql = $db->prepare("SELECT artists.nameArtist FROM simpletab.artists");
            $sql->bindParam(':nameArtist', $nameArtist, PDO::PARAM_STR);
            $sql->execute();
            $result = $sql->fetchAll();
            return $result;
        }
        catch (PDOException $e) {
            return false;
        }
    }
```

```php

    /**
     * Ajoute un artiste en base depuis son nom
     * @param $nameArtist
     * @return true si réussi sinon false
     */
    function addArtist($nameArtist)
    {
        $db = Database::getInstance();

        try {
            $sql = $db->prepare("INSERT INTO simpletab.artists (nameArtist) VALUES ( :nameArtist);");
            $sql->bindParam(':nameArtist', $nameArtist, PDO::PARAM_STR);
            $sql->execute();
            return true;
        }
        catch (PDOException $e) {
            return false;
        }
    }

    /**
     * Récupère un artiste par son nom
     * @param $nameArtist
     * @return l'artiste si résussi sinon false
     */
    function getArtistByName($nameArtist)
    {
        $db = Database::getInstance();

        try {
            $sql = $db->prepare("SELECT * FROM simpletab.artists WHERE artists.nameArtist = :nameArtist;");
            $sql->bindParam(':nameArtist', $nameArtist, PDO::PARAM_STR);
            $sql->execute();
            $result = $sql->fetchAll();
            return $result;
        }
        catch (PDOException $e) {
            return false;
        }
    }

    /**
     * @var Tableau des artistes
     */
    private $artist;
}
```

1.3.6 rate.php

```php
<?php
/**
 * Auteur: romain.ssr@eduge.ch
 * Date : 22.05.2018
 * projet: SimpleTab
 */

/**
 * Classe des artistes
 * @author RS
 */
class rate implements JsonSerializable
{
    /**
     * Rate constructor.
     * @param int $InArtistId -> L'id de l'artiste
     * @param string $InArtistName -> Le nom de l'artiste
     */
    public function __construct ($InRateId = - 1, $InRate = -1, $InRateIdTab = -1, $InRateIdUser =-1)
    {
        $this->id = $InRateId;
        $this->rate = $InRate;
        $this->idTab = $InRateIdTab;
        $this->idUser = $InRateIdUser;
    }

    /**
     * @brief On ne laisse pas cloner une tablature
     */
    private function __clone ()
    {}

    /**
     * @brief Est-ce que cet objet est valide
     *
     * @return True si valide, autrement false
     */
    public function isValid ()
    {
```

```
40          return ( $this->id == -1 || $this->rate = -1 || $this->idTab = -1 || $this->idUser = -1) ? false : ↩
        true;
41      }
42
43      /**
44       * @brief Getter
45       *
46       * @return L'identifiant de la note
47       */
48      public function getId ()
49      {
50          return $this->id;
51      }
52
53      /**
54       * @brief Getter
55       *
56       * @return La valeur de la note
57       */
58      public function getRate ()
59      {
60          return $this->rate;
61      }
62
63      /**
64       * @brief Getter
65       *
66       * @return L'id de la tablature associée
67       */
68      public function getIdTab ()
69      {
70          return $this->idTab;
71      }
72
73      /**
74       * @brief Getter
75       *
76       * @return L'id de l'utilisateur associé
77       */
78      public function getIdUser ()
79      {
80          return $this->idUser;
81      }
82
83
84      public function jsonSerialize()
85      {
86          return get_object_vars($this);
87      }
88
89      /** Identifiant de la note */
90      private $id;
91      /** La valeur de la note */
92      private $rate;
93      /** L'id de la tablature associée */
94      private $idTab;
95      /** L'id de l'utilisateur associé */
96      private $idUser;
97  }
```

1.3.7 tablature.php

```
1   <?php
2   /**
3    * Auteur: romain.ssr@eduge.ch
4    * Date : 15.05.2018
5    * projet: SimpleTab
6    */
7
8   /**
9    * Classe des tablatures
10   * @author RS
11   */
12  class Tablature implements JsonSerializable
13  {
14      /**
15       * @brief Class Constructor
16       * @param $InTabId L'identifiant de la tablature
17       * @param $InTabTitle Le titre de la tablature
18       * @param $InTabPath Le chemin de la tablature au format XML
19       * @param $InTabRate La note de la tablature sur 5
20       * @param $InLvlTab Le niveau de la tablature (0 -> facile; 1-> moyen; 2-> difficile)
21       * @param $InUserID L'id de l'utilisateur qui a posté la tablature
22       * @param $InArtistID L'id de l'artiste qui a créé la tablature
23
24       */
25      public function __construct ($InTabId = - 1, $InTabTitle = "", $InTabPath = "", $InTabRate = -1, ↩
        $InLvlTab = -1, $InUserID = - 1,$InArtistID = - 1)
26      {
27          $this->id= $InTabId;
```

```php
        $this->title = $InTabTitle;
        $this->path = $InTabPath;
        $this->rate = $InTabRate;
        $this->lvl = $InLvlTab;
        $this->userId = $InUserID;
        $this->artistId = $InArtistID;
    }

    /**
     * @brief On ne laisse pas cloner une tablature
     */
    private function __clone ()
    {}

    /**
     * @brief Est-ce que cet objet est valide
     *
     * @return True si valide, autrement false
     */
    public function isValid ()
    {
        return ( $this->id == -1 || $this->title == "" || $this->path == "" || $this->rate == -1 || $this->lvl == -1 || $this->userId == -1 || $this->artistId == -1) ? false : true;
    }

    /**
     * @brief Getter
     *
     * @return L'identifiant de la tablature
     */
    public function getId ()
    {
        return $this->id;
    }

    /**
     * @brief Getter
     *
     * @return Le titre de la tablature
     */
    public function getTitle ()
    {
        return $this->title;
    }

    /**
     * @brief Getter
     *
     * @return Le chemin de la tablature
     */
    public function getPath ()
    {
        return $this->path;
    }

    /**
     * @brief Getter
     *
     * @return La note de la tablature
     */
    public function getRate ()
    {
        return $this->rate;
    }

    /**
     * @brief Getter
     *
     * @return Le niveau de la tablature
     */
    public function getLvl ()
    {
        return $this->lvl;
    }

    /**
     * @brief Getter
     *
     * @return L'identifiant de l'utilisateur qui a posté la tablature
     */
    public function getUserId ()
    {
        return $this->userId;
    }

    /**
     * @brief Getter
     *
     * @return L'identifiant de l'artiste qui a créé la tablature
     */
    public function getArtistId ()
    {
        return $this->artistId;
    }
```

```
121
122     /**
123      * Sérialize l'objet en json
124      * @return array|mixed
125      */
126     public function jsonSerialize()
127     {
128         return get_object_vars($this);
129     }
130
131
132     /** Identifiant de la tablature */
133     private $id;
134     /** Le titre de la tablature */
135     private $title;
136     /** Le chemin de la tablature au format XML */
137     private $path;
138     /** La note de la tablature sur 5 */
139     private $rate;
140     /** Le niveau de la tablature (0 -> facile; 1-> moyen; 2-> difficile)*/
141     private $lvl;
142     /** L'id de l'utilisateur qui a posté la tablature */
143     private $userId;
144     /** L'id de l'artiste qui a créé la tablature */
145     private $artistId;
146 }
```

1.3.8 user.php

```
1  <?php
2  /**
3   * Auteur: romain.ssr@eduge.ch
4   * Date : 15.05.2018
5   * projet: SimpleTab
6   */
7
8  /**
9   * Classe des utilisateurs
10  * @author RS
11  */
12 class Users implements JsonSerializable
13 {
14     /**
15      * @brief Class Constructor
16      * @param $InUserId L'identifiant de l'utilisateur
17      * @param $InUserName Le nom de l'utilisateur
18      * @param $InUserForename Le prénom de l'utilisateur
19      * @param $InUserPassword Le mot de passe de l'utilisateur
20      * @param $InUserEmail L'email de l'utilisateur
21      * @param $InUserPseudo Le pseudonyme de l'utilisateur
22      */
23     public function __construct ($InUserId = - 1, $InUserName = "", $InUserForename = "", $InUserPassword = ↩
        "", $InUserEmail = "", $InUserPseudo = "")
24     {
25         $this->id= $InUserId;
26         $this->name = $InUserName;
27         $this->forename = $InUserForename;
28         $this->password = $InUserPassword;
29         $this->email = $InUserEmail;
30         $this->pseudo = $InUserPseudo;
31
32     }
33
34     /**
35      * @brief On ne laisse pas cloner un utilisateur
36      */
37     private function __clone ()
38     {}
39
40     /**
41      * @brief Est-ce que cet objet est valide
42      *
43      * @return True si valide, autrement false
44      */
45     public function isValid ()
46     {
47         return ( $this->id == -1 || $this->name == "" || $this->forename == "" || $this->password == "" || ↩
        $this->email == "" || $this->pseudo == "") ? false : true;
48     }
49
50     /**
51      * @brief Getter
52      *
53      * @return L'identifiant de l'utilisateur
54      */
55     public function getId ()
56     {
57         return $this->id;
58     }
59
```

```php
    /**
     * @brief Getter
     *
     * @return Le nom de l'utilisateur
     */
    public function getName ()
    {
        return $this->name;
    }

    /**
     * @brief Getter
     *
     * @return Le prénom de l'utilisateur
     */
    public function getForename ()
    {
        return $this->forename;
    }

    /**
     * @brief Getter
     *
     * @return Le mot de passe de l'utilisateur
     */
    public function getPassword ()
    {
        return $this->password;
    }

    /**
     * @brief Getter
     *
     * @return L'email de l'utilisateur
     */
    public function getEmail()
    {
        return $this->email;
    }

    /**
     * @brief Getter
     *
     * @return Le pseudo de l'utilisateur
     */
    public function getPseudo ()
    {
        return $this->pseudo;
    }

    /**
     * Sérialize l'objet en json
     * @return array|mixed
     */
    public function jsonSerialize()
    {
        return get_object_vars($this);
    }


    /** L'identifiant de l'utilisateur */
    private $id;
    /** Le nom de l'utilisateur */
    private $name;
    /** Le prénom de l'utilisateur */
    private $forename;
    /** Le mot de passe de l'utilisateur */
    private $password;
    /** L'email de l'utilisateur */
    private $email;
    /** Le pseudo de l'utilisateur */
    private $pseudo;

}
?>
```

1.3.9 comment.php

```php
<?php
/**
 * Auteur: romain.ssr@eduge.ch
 * Date : 17.05.2018
 * projet: SimpleTab
 */

/**
 * Classe des commentaires
 * @author RS
 */
class Comment implements JsonSerializable
```

```
13 {
14     /**
15      * @brief Class Constructor
16      * @param $InCommentId L'identifiant du commentaire
17      * @param $InCommentContent Le contenu du commentaire
18      * @param $InTabId L'id de la tablature associée au commentaire
19      * @param $InUserId L'id de l'utilisateur associée au commentaire
20      */
21     public function __construct ($InCommentId = -1, $InCommentContent = "", $InTabId = "",  $InUserId = - 1)
22     {
23         $this->id= $InCommentId;
24         $this->commentContent = $InCommentContent;
25         $this->tabId = $InTabId;
26         $this->userId = $InUserId;
27     }
28
29     /**
30      * @brief On ne laisse pas cloner une tablature
31      */
32     private function __clone ()
33     {}
34
35     /**
36      * @brief Est-ce que cet objet est valide
37      *
38      * @return True si valide, autrement false
39      */
40     public function isValid ()
41     {
42         return ( $this->id == -1 || $this->commentContent == "" || $this->tabId == -1 ||  $this->userId == ↩
    -1 ) ? false : true;
43     }
44
45     /**
46      * @brief Getter
47      *
48      * @return L'identifiant du commentaire
49      */
50     public function getId ()
51     {
52         return $this->id;
53     }
54
55     /**
56      * @brief Getter
57      *
58      * @return Le contenu du commentaire
59      */
60     public function getContent ()
61     {
62         return $this->commentContent;
63     }
64
65     /**
66      * @brief Getter
67      *
68      * @return L'id de la tablature associée au commentaire
69      */
70     public function getTabId ()
71     {
72         return $this->tabId;
73     }
74
75     /**
76      * @brief Getter
77      *
78      * @return L'id de l'utilisateur associée au commentaire
79      */
80     public function getUserId ()
81     {
82         return $this->userId;
83     }
84
85     /**
86      * Serialize l'objet en json
87      * @return array|mixed
88      */
89     public function jsonSerialize()
90     {
91         return get_object_vars($this);
92     }
93
94
95     /** Identifiant du commentaire */
96     private $id;
97     /** Le contenu du commentaire */
98     private $commentContent;
99     /** L'id de la tablature associée au commentaire */
100    private $tabId;
101    /** L'id de l'utilisateur associé au commentaire */
102    private $userId;
103
104 }
```

1.3.10 Artist.php

```php
<?php
/**
 * Auteur: romain.ssr@eduge.ch
 * Date : 18.05.2018
 * projet: SimpleTab
 */


/**
 * Classe des artistes
 * @author RS
 */
class Artist implements JsonSerializable
{
    /**
     * Artist constructor.
     * @param int $InArtistId -> L'id de l'artiste
     * @param string $InArtistName -> Le nom de l'artiste
     */
    public function __construct ($InArtistId = - 1, $InArtistName = "")
    {
        $this->id= $InArtistId;
        $this->name = $InArtistName;
    }

    /**
     * @brief On ne laisse pas cloner une tablature
     */
    private function __clone ()
    {}

    /**
     * @brief Est-ce que cet objet est valide
     *
     * @return True si valide, autrement false
     */
    public function isValid ()
    {
        return ( $this->id == -1 || $this->name == "") ? false : true;
    }

    /**
     * @brief Getter
     *
     * @return L'identifiant de l'artiste
     */
    public function getId ()
    {
        return $this->id;
    }

    /**
     * @brief Getter
     *
     * @return Le nom de l'artiste
     */
    public function getName ()
    {
        return $this->name;
    }


    public function jsonSerialize()
    {
        return get_object_vars($this);
    }

    /** Identifiant de l'artiste */
    private $id;
    /** Le nom de l'artiste */
    private $name;

}
```

1.3.11 database.php

```php
<?php
/**
 * @author  dominique.aigroz@edu.ge.ch
 */
require_once '../config/conparam.php';

/**
 * @brief Helper class encapsulating
 *      the PDO object
 * @author  dominique.aigroz@kadeo.net
 * @remark
```

```
 */

class Database
{
  private static $pdoInstance;
  /**
   * @brief Class Constructor - Create a new database connection if one doesn't exist
  Â *       Set to private so no-one can create a new instance via ' = new KDatabase();'
  Â */
  private function __construct() {}
  /**
   * @brief Like the constructor, we make __clone private so nobody can clone the instance
  Â */
  private function __clone() {}
  /**
   * @brief Returns DB instance or create initial connection
   * @return $objInstance;
   */
  public static function getInstance() {
    if(!self::$pdoInstance){
      try{

        $dsn = EDB_DBTYPE.':host='.EDB_HOST.';port='.EDB_PORT.';dbname='.EDB_DBNAME;
          self::$pdoInstance = new PDO($dsn, EDB_USER, EDB_PASS, array('charset'=>'utf8'));
        self::$pdoInstance->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
      }catch(PDOExceptionÂ  $e ){
        echo "KDatabase Error: ".$e->getMessage();
      }
    }
    return self::$pdoInstance;
  } # end method
  /**
   * @brief Passes on any static calls to this class onto the singleton PDO instance
   * @param   $chrMethod    The method to call
   * @param   $arrArguments The method's parameters
   * @return  $mix      The method's return value
   */
  final public static function __callStatic( $chrMethod, $arrArguments ) {
    $pdo = self::getInstance();
    return call_user_func_array(array($pdo, $chrMethod), $arrArguments);
  } # end method
}
```

1.4 public

1.4.1 css

style.css

```
/**
 * Auteur: romain.ssr@eduge.ch
 * Date : 18.05.2018
 * projet: SimpleTab
 * description: feuille de style
 */
.artistName{
color : darkred;
}
a:hover {
    text-decoration: underline;
    cursor: pointer;
}
```

1.4.2 js

function.js

```
/**
 * @copyright dominique.aigroz@edu.ge.ch dario.gng@eduge.ch romain.ssr@eduge.ch
 * Effectue un appel ajax et exécute une fonction en cas de succès
 * @param string url        Le chemin vers le fichier qui va récupérer nos données
 * @param string callback   Le nom de la fonction à  exécuter après avoir reçu les données. Il ne faut pas ↩
    mettre de parenthèses
 * @param object params     Object qui contient tout les paramètres à  envoyer
 * @param boolean async   Pour activer / désactiver l'asynchrone d'un appel ajax
 */

function get_data(url, callback, params = {}, async) {
    // Copier l'objet params dans une variable locale
    // qui sera simplement utilisées pour l'appel Ajax
    var dp = params;
```

```
    // On utilise le paramètre de la fonction qui est stocké sur le stack
    // pour créer un tableau qui contient les paramètres additionnels qui se
    // trouvent après params.
    // Si on stocke ces paramètres dans un tableau créé dans cette function,
    // il sera détruit après l'appel Ajax et on aura plus rien lorsqu'on sera
    // rappelé de manière asynchrone.
    params = Array();
    for (var i = 4; i < arguments.length; i++){
        params.push(arguments[i]);
    }
    $.ajax({
        method: 'POST',
        url: url,
        data: dp,
        dataType: 'json',
        async : async,
        success: function (data) {
            var msg = '';

            switch (data.ReturnCode){
                case 0 : // tout bon
                    // On récupère par params, notre tableau des arguments.
                    params.unshift(data.Data);
                    callback.apply(this, params);
                    break;
                case 1: // erreur param
                    alert("Veuillez remplir tous les champs correctement");
                    break;
                case 2 : // problème récup données
                    alert("Un problème est survenu");
                case 3 : // problème encodage sur serveur
                    alert("Un problème est survenu");

                default:

                    break;
            }
        },
        error: function(jqXHR){
            // Votre gestion de l'erreur
        }
    });
}

/**
 * Retourne le niveau de la tablature en toute lettres : 0 -> facile ;  1 -> moyen ; 2 -> difficile
 * @param $lvlNumber -> le niveau de la tablature en chiffre
 */
function getDifficultyInLetters($lvlNumber)
{
    switch ($lvlNumber)
    {
        case "0":
            return "Facile";
            break;
        case "1":
            return "Moyen";
            break;

        case "2":
            return "Difficile";
            break;

    }
}

/**
 * Envoie l'id d'une tablature à  la fonction getInfoTab
 * @param idTab
 */
function passTabIdToGetInfosTab(idTab)
{
    get_data("../controller/getInfosTab.php",getInfosTab,{'idTab' : idTab},true);
    function getInfosTab(data)
    {
        if(data != false)
        {
            if($.isEmptyObject(data.metadata.capo))
            {
                data.metadata.capo = "";
            }
            if($.isEmptyObject(data.metadata.key))
            {
                data.metadata.key = "";
            }
            lvlNumber = getDifficultyInNumber(data.metadata.lvl);
            $('#modifyTitle').val(data.metadata.title);
            $('#modifyAuthor').val(data.metadata.author);
            $('#modifyLvl select').val(data.metadata.level);
            $('#modifyCapo').val(data.metadata.capo);
            $('#modifyKey').val(data.metadata.key);
            $('#modifyTuning').val(data.metadata.tuning);
            $('#modifyTablatureBody').val(data.corpse);

```

```
108         }
109     }
110 }
111 
112 /**
113  * Supprime la tablature par son id et rafraichît le tableau des tablatures de l'utilisateur
114  * @param idTab
115  * @param idUser
116  */
117 function deleteTabById(idTab, idUser)
118 {
119     if(confirm("Voulez-vous vraiment supprimer la tablature ?")) {
120         get_data("../controller/deleteTabById.php", deleteTabById, {'idTab': idTab}, true);
121 
122         function deleteTabById(data) {
123             var message = "";
124             if (data == true) {
125                 message = "<div class=\"alert alert-success text-center\" role=\"alert\">" +
126                     " Votre tablature a bien été suprimée.</div>";
127             }
128             else {
129                 message = "<div class=\"alert alert-danger text-center\" role=\"alert\">" +
130                     "Un problème est survenu </div>";
131             }
132             $('#message').append(message);
133             alert('la tablature a été bien supprimée');
134             reloadTabByUSers(idUser);
135 
136         }
137     }
138 }
139 
140 /**
141  * Quand l'adinistrateur refuse une tablature, elle est supprimée ainsi que la tablature au format XML
142  * @param idTab
143  */
144 function refuseTab(idTab)
145 {
146     get_data("../controller/deleteTabById.php",deleteTabById,{'idTab' : idTab},true);
147     function deleteTabById(data)
148     {
149         var message = "";
150         if (data == true) {
151             alert('la tablature a été refusée');
152             getAllNonApprouvedTabs();
153 
154         }
155         else {
156             alert('un problème est survenu');
157 
158         }
159 
160     }
161 }
162 
163 /**
164  * Quand l'adinistrateur accepte une tablature, son champ approuved passe à  1. La tablature devient ↩
        consultable par tous.
165  * @param idTab
166  */
167 function accepTab(idTab)
168 {
169     get_data("../controller/approuveTab.php",approuveTab,{'idTab' : idTab},true);
170     function approuveTab(data)
171     {
172         var message = "";
173         if (data == true)
174         {
175             alert('la tablature a été approuvée');
176             getAllNonApprouvedTabs();
177 
178         }
179         else
180         {
181             alert('un problème est survenu');
182         }
183 
184     }
185 }
186 
187 /**
188  * rafraichît le tableau des tablatures postées par l'utilisateurs par leur ID
189  * @param idUser
190  */
191 function reloadTabByUSers(idUser)
192 {
193     ↩
      get_data("../controller/getTabAndRelatedArtistPostedByUser.php",getTabAndRelatedArtistPostedByUser,{'idUser' ↩
      : idUser},true);
194     function getTabAndRelatedArtistPostedByUser(data) {
195         $('#tabs').empty();
196         data.forEach(function (tablature) {
197             var artist = $('<a class="artistName" >' + tablature.nameArtist + '</a>');
198             var td = $("<td>").append(artist);
```

```
            var lvl = getDifficultyInLetters(tablature.lvlTab);
            var tr = $("<tr>").append(td);
            $(tr).append(
                '<td>' + tablature.titleTab + '</td>' +
                '<td>' + lvl + '</td>' +
                '<td>' + tablature.rateTab + '</td>' +
                '<td style="width:1%;"><button  class="btn btn-dark border-0" style="background-color: #20262b;" onclick="deleteTabById(' + tablature.idTab + ','+idUser+')"><i class="fas fa-trash-alt"></i></button></td>' +
                '<td style="width:1%;"><button class="btn btn-dark border-0" data-toggle="modal" data-target="#modifyTab" style="background-color: #20262b;" onclick="passTabIdToGetInfosTab(' + tablature.idTab + ')"><i class="fas fa-pencil-alt"></i></button></td>' +
                '</tr>');
            $('#tabs').append(tr);


            $(artist).click(function () {
                var artistName = $(this).text();
                get_data("../controller/getTabByArtist.php", getTabByArtist, {'artistName': artistName}, true);

                function getTabByArtist(data) {
                    $('#tabs').empty();
                    data.forEach(function (tablature) {
                        var $lvl = getDifficultyInLetters(tablature.lvlTab);
                        var $row = $('<tr>' +
                            '<td><a class="artistName" >' + tablature.nameArtist + '</a></td>' +
                            '<td>' + tablature.titleTab + '</td>' +
                            '<td>' + $lvl + '</td>' +
                            '<td>' + tablature.rateTab + '</td>' +
                            '</tr>');
                        $('#tabs').append($row);

                    });
                }
            });


        });
    }
}

/**
 * Récupère toutes les tablatures non approuvée (champs approuved = 0)
 */
function getAllNonApprouvedTabs()
{
    get_data("../controller/getAllNonApprouvedTab.php",getAllNonApprouvedTab,{},true);
    function getAllNonApprouvedTab(data) {
        $('#tabs').empty();
        data.forEach(function (tablature) {
            var artist = tablature.nameArtist;
            var td = $("<td>").append(artist);
            var lvl = getDifficultyInLetters(tablature.lvlTab);
            var tr = $("<tr>").append(td);
            $(tr).append(
                '<td><a class="titleTab">'+tablature.titleTab+'</a></td>' +
                '<td>' + lvl + '</td>' +
                '<td>' + tablature.rateTab + '</td>' +
                '<td style="width:1%;"><button  class="btn btn-dark border-0" style="background-color: #20262b;" onclick="accepTab(' + tablature.idTab +')"><i class="fas fa-check"></i></button></td>' +
                '<td style="width:1%;"><button class="btn btn-dark border-0"  style="background-color: #20262b;" onclick="refuseTab(' + tablature.idTab + ')"><i class="fas fa-times"></i></button></td>' +
                '</tr>');
            $('#tabs').append(tr);


            $('.titleTab').click(function () {
                var titleTab = $(this).text();
                get_data("../controller/getTabByTitle.php",getTabByTitle,{'titleTab':titleTab},true);
                function getTabByTitle(data) {
                    data.forEach(function(tablature) {
                        window.location.href= "../view/tablaturePage.php?idTab="+tablature.idTab;
                    });
                }
            });
        });
    }
}

/**
 * Récupère les utilisateurs et le nombre de tablatures qu'ils ont posté
 */
function getUsersAndNbTabPosted()
{
    get_data("../controller/getUsersAndNbTabPosted.php",getUsersAndNbTabPosted,{},true);
    function getUsersAndNbTabPosted(data) {
        $('#users').empty();
        data.forEach(function (user) {
            var pseudo =  user.pseudoUser;
            var email = user.emailUser;
            var nbTab = user.nbTab;
            var tr = $("<tr>");
            $(tr).append(
```

```
286                 '<td>' + pseudo + '</td>' +
287                 '<td>' + email + '</td>' +
288                 '<td>' + nbTab + '</td>' +
289                 '<td style="width:1%;"><button class="btn btn-dark border-0"  style="background-color: ↩
    #20262b;" onclick="deleteUserById(' + user.idUsers + ')"><i class="fas ↩
    fa-trash-alt"></i></button></td>' +
290                 '</tr>');
291             $('#users').append(tr);
292         });
293     }
294 }
295
296 /**
297  * Supprime la tablatures à  partir d'un ID
298  * @param idUser
299  */
300 function deleteUserById(idUser)
301 {
302     if(confirm("Voulez-vous vraiment supprimer l'utilisateur ?")) {
303         get_data("../controller/deleteUserById.php", deleteUserById, {'idUser': idUser}, true);
304
305         function deleteUserById(data) {
306             var message = "";
307             if (data == true) {
308                 alert("l'utilisateur a été supprimé avec succès");
309                 getAllNonApprouvedTabs();
310                 getUsersAndNbTabPosted();
311
312             }
313             else {
314                 alert('un problème est survenu');
315
316             }
317
318         }
319     }
320 }
321
322 function updateTabRate()
323 {
324     get_data("../controller/deleteUserById.php",deleteUserById,{'idUser' : idUser},true);
325
326 }
327
328 /**
329  * Renvoie la difficulté d'une tablature en toute lettres depuis un chiffre
330  * @param $lvlInLetter
331  * @returns le chiffre correspondant à  la difficulté
332  */
333 function getDifficultyInNumber($lvlInLetter)
334 {
335     $lvl = $.trim($lvlInLetter);
336     switch ($lvl)
337     {
338         case "Facile":
339             return 0;
340             break;
341         case "Moyen":
342             return 1;
343             break;
344         case "Difficile":
345             return 2;
346             break;
347     }
348
349 }
350
351 function identifyUser(pseuoOrMail,password)
352 {
353
354     get_data("../controller/identifyUser.php",identifyUser,{'mailOrPseudo' :pseuoOrMail, 'pwdConnexion' : ↩
     password},true);
355     function identifyUser(data) {
356         if(data.length != 0)
357         {
358             data.forEach(function(user) {
359                 location.reload();
360             });
361         }
362         else
363         {
364             alert("l'utilisateur n'existe pas ou le mot de passe est incorrect")
365             get_data("../controller/destroySession.php",destroySession,{},true);
366             function destroySession(data)
367             {
368
369             }
370         }
371
372     }
373 }
```

1.5 tabs

1.5.1 81.php

```
<?php
$xmlstr = <<<XML
<?xml version = "1.0" encoding="UTF-8" standalone="yes" ?>
<tabs>
     <metadata>
            <title> Hallelujah </title>
            <author>Jeff Buckley</author>
            <tuning>E A D G B E</tuning>
            <capo></capo>
            <key></key>
            <level> Facile</level>
     </metadata>
     <corpse>
     [Intro]
C Am C Am


[Verse 1]
  C                 Am
I heard there was a secret chord
     C                   Am
That David played and it pleased the lord
    F                G               C        G
But you don't really care for music, do you?
        C                 F           G
Well it goes like this the fourth, the fifth
    Am                F
The minor fall and the major lift
    G            E7             Am
The baffled king composing hallelujah


[Chorus]
     F           Am          F           C    G   C      G
Hallelujah, hallelujah, hallelujah, hallelu-u-u-u-jah ....


[Verse 2]
          C                        Am
Well your faith was strong but you needed proof
    C               Am
You saw her bathing on the roof
    F              G             C            G
Her beauty and the moonlight overthrew you
C                   F       G
She tied you to her kitchen chair
    Am                       F
She broke your throne and she cut your hair
    G                   E7            Am
And from your lips she drew the hallelujah


[Chorus]
     F           Am          F           C    G   C      G
Hallelujah, hallelujah, hallelujah, hallelu-u-u-u-jah ....


[Verse 3]
C              Am
Baby I've been here before
     C                     Am
I've seen this room and I've walked this floor
  F             G             C          G
I used to live alone before I knew you
C                           F       G
I've seen your flag on the marble arch
    Am                  F
But love is not a victory march
       G                E7           Am
It's a cold and it's a broken hallelujah


[Chorus]
     F           Am          F           C    G   C      G
Hallelujah, hallelujah, hallelujah, hallelu-u-u-u-jah ....


[Verse 4]
     C                        Am
Well there was a time when you let me know
       C             Am
What's really going on below
    F             G               C        G
But now you never show that to me do you
      C             F        G
But remember when I moved in you
        Am            F
And the holy dove was moving too
```

```
    G               E7            Am
And every breath we drew was hallelujah


[Chorus]
     F           Am          F           C    G   C      G
Hallelujah, hallelujah, hallelujah, hallelu-u-u-u-jah ....


[Verse 5]
      C              Am
Well, maybe there's a god above
    C            Am
But all I've ever learned from love
    F               G          C          G
Was how to shoot somebody who outdrew you
     C                  F       G
It's not a cry that you hear at night
     Am                 F
It's not somebody who's seen the light
       G               E7          Am
It's a cold and it's a broken hallelujah


[Outro]
     F           Am          F           C    G   C      G
Hallelujah, hallelujah, hallelujah, hallelu-u-u-u-jah ....
     </corpse>
</tabs>
XML;
?>
```

1.5.2 83.php

```
<?php
$xmlstr = <<<XML
<?xml version = "1.0" encoding="UTF-8" standalone="yes" ?>
<tabs>
     <metadata>
            <title> Perfect </title>
            <author>Ed Sheeran</author>
            <tuning>E A D G B E</tuning>
            <capo></capo>
            <key></key>
            <level> Facile</level>
     </metadata>
     <corpse>
     Play this song with the regular shapes of the chords given in this song
or use the ones that can be heard on the recording:

     E-A-D-G-B-e
G    3-x-0-0-3(3)
Em   0-2-2-0-3(3) (= Em7)
C    x-3-2-0-3(3) (= Cadd9)

[Intro]
G

[Verse]
          G        Em
I found a love for me
             C                          D
Darling just dive right in, and follow my lead
               G           Em
Well I found a girl beautiful and sweet
         C                                D
I never knew you were the someone waiting for me

[Pre-Chorus]
                             G
Cause we were just kids when we fell in love
            Em                  C               G  D
Not knowing what it was, I will not give you up this ti-ime
                  G                         Em
But darling just kiss me slow, your heart is all I own
            C                   D
And in your eyes you're holding mine

[Chorus]
      Em   C             G          D              Em
Baby, I'm dancing in the dark, with you between my arms
C               G     D               Em
Barefoot on the grass, listening to our favorite song
          C               G                D              Em
When you said you looked a mess, I whispered underneath my breath
         C               G        D             G
But you heard it, darling you look perfect tonight

| G D/F# Em D | C  D  |

```

```
[Verse]
               G                     Em
Well I found a woman, stronger than anyone I know
                C                                          D
She shares my dreams, I hope that someday I'll share her home
           G             Em
I found a love, to carry more than just my secrets
         C                              D
To carry love, to carry children of our own

[Pre-Chorus]
                              G                     Em
We are still kids, but we're so in love, fighting against all odds
                  C              G  D
I know that we'll be alright this ti-ime
               G                             Em
Darling just hold my hand, be my girl, I'll be your man
          C              D
I see my future in your eyes

[Chorus]
      Em   C              G         D              Em
Baby, I'm dancing in the dark, with you between my arms
C                G     D                Em
Barefoot on the grass, listening to our favorite song
        C                 G                D
When I saw you in that dress, looking so beautiful
  Em       C                    G        D          [ G ]
I don't deserve this, darling you look perfect tonight

[Interlude]
| G  | % | Em | % |
| C  | % | D  | % |

[Chorus]
      Em   C              G         D              Em
Baby, I'm dancing in the dark, with you between my arms
C                G     D                Em
Barefoot on the grass, listening to our favorite song
         C                 G                D            Em
I have faith in what I see, now I know I have met an angel
    C          G          D
In person, and she looks perfect

[Outro]
  G/B     C           Dadd4    D        [ G ]
I don't deserve this, you look perfect tonight

| G D/F# Em D | C  D  | G
     </corpse>
</tabs>
XML;
?>
```

1.5.3 84.php

```
<?php
$xmlstr = <<<XML
<?xml version = "1.0" encoding="UTF-8" standalone="yes" ?>
<tabs>
     <metadata>
            <title> Photograph </title>
            <author>Ed Sheeran</author>
            <tuning>E A D G B E</tuning>
            <capo></capo>
            <key></key>
            <level> Facile</level>
     </metadata>
     <corpse>
     [Intro]

D   Bm   A   G


[Verse]

           D
Loving can hurt
           Bm
Loving can hurt sometimes
             A                          G
But it's the only thing that I know
             D
When it gets hard
                   Bm
You know it can get hard sometimes
          A                                G
It is the only thing that makes us feel alive


```

```
[Pre-Chorus]

Bm                      G
We keep this love in a photograph
D                       A
We make these memories for ourselves
          Bm
Where our eyes are never closing
     G
Our hearts were never broken
     D                       A
And times forever frozen still


[Chorus]

           D
So you can keep me inside the pocket of your
A
Ripped jeans holding me closer till our
Bm                      G
Eyes meet, you won't ever be alone
                         D
Wait for me to come home


[Verse]

            D
Loving can heal
           Bm
Loving can mend your soul
             A                        G
And it's the only thing that I know
                    D
I swear it will get easier
                            Bm
remember that with every piece of ya
             A                                  G
It is the only thing we take with us when we die


[Pre-Chorus]

Bm                      G
We keep this love in a photograph
D                       A
We make these memories for ourselves
          Bm
Where our eyes are never closing
     G
Our hearts were never broken
     D                      A
And times forever frozen still


[Chorus]

           D
So you can keep me inside the pocket of your
A
Ripped jeans holding me closer till our
Bm                        G
Eyes meet, you won't ever be alone


[Bridge]

           D
And if you hurt me that's ok baby, only
A
Words bleed inside these pages you just
Bm                       G
Hold me and I won't ever let you go


[Interlude]

                     Bm
Wait for me to come home
                     G
Wait for me to come home
                     D
Wait for me to come home
                     A
Wait for me to come home


[Outro]

           D
Oh you can fit me inside the necklace you got when you were
A
16 next to your heartbeat where I
```

```
Bm                                  G
Should be, keep it deep within your soul
            D
And if you hurt me that's ok baby only
A
Words bleed inside these pages you just
Bm                                  G
Hold me and I won't ever let you go
            D
When I'm away I will remember how you
A                                        Bm
Kissed me under the lamp post back on sixth street
                                    G
Hearing you whisper through the phone
                        D
Wait for me to come home



     </corpse>
</tabs>
XML;
?>
```

1.6 view

1.6.1 homePage.php

```
<?php
/**
 * Auteur: romain.ssr@eduge.ch
 * Date : 09.05.2018
 * projet: SimpleTab
 * description : page d'accueil du site
 */

session_start();

require_once"modals.php";

$navIdOrButton="";
$navMenu1 = "";
$navMenu2 = "";

if(isset($_SESSION['user']))
{
    $navIdOrButton = "<div class=\"dropdown\">
                            <button type=\"button\" onclick=\" \" class=\"btn btn-light dropdown-toggle\" data-toggle=\"dropdown\" aria-haspopup=\"true\" aria-expanded=\"false\"> ".$_SESSION['user'][0]['pseudoUser']."</button>
                            <div class=\"dropdown-menu\" aria-labelledby=\"dropdownMenuButton\">
                                <a class=\"dropdown-item\" href=\"../controller/destroySession.php\">Déconnexion</a>
                            </div>
                        </div>";
    if($_SESSION['user'][0]['role_idrole'] == 0)
    {
        $navMenu1 = "<h5><a class=\"nav-link\" href=\"../view/homePage.php\">Accueil <span class=\"sr-only\">(current)</span></a></h5>";
        $navMenu2 = "<h5><a class=\"nav-link\" href=\"../view/tablatureManagerPage.php\">Gestion des tablatures </a></h5>";
    }
    elseif($_SESSION['user'][0]['role_idrole'] == 1)
    {
        $navMenu1 ="<h5><a class=\"nav-link\" href=\"../view/homePage.php\">Accueil <span class=\"sr-only\">(current)</span></a></h5>";
        $navMenu2 = "<h5><a class=\"nav-link\" href=\"../view/tablatureAndUserManagerPage.php\">Gestion des tablatures et utilisateurs </a></h5>";
    }
}
else
{
    $navIdOrButton = "<button type=\"button\" class=\"btn btn-light\" data-toggle=\"modal\" data-target=\"#addUser\">S'inscrire</button> | <button type=\"button\" class=\"btn btn-light\"data-toggle=\"modal\" data-target=\"#connectUser\">S'identifier</button>";
    $navMenu1 = "<h5><a class=\"nav-link\" href=\"../view/homePage.php\">Accueil <span class=\"sr-only\">(current)</span></a></h5>";
}
?>
<! DOCTYPE HTML>
<html>
<head>
    <title>Accueil</title>
    <script src="https://code.jquery.com/jquery-3.3.1.js" integrity="sha256-2Kok7MbOyxpgUVvAk/HJ2jigOSYS2auK4Pfzbm7uH60=" crossorigin="anonymous"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js" integrity="sha384-ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/l8WvCWPIPm49" crossorigin="anonymous"></script>
```

```
48      <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.1/js/bootstrap.min.js" ↩
         integrity="sha384-smHYKdLADwkXOn1EmN1qk/HfnUcbVRZyYmZ4qpPea6sjB/pTJ0euyQp0Mk8ck+5T" ↩
         crossorigin="anonymous"></script>
49      <link rel="stylesheet" type="text/css" href="../public/css/style.css">
50      <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.0/css/bootstrap.min.css" ↩
         integrity="sha384-9gVQ4dYFwwWSjIDZnLEWnxCjeSWFphJiwGPXr1jddIhOegiu1FwO5qRGvFXOdJZ4" ↩
         crossorigin="anonymous">
51
52      <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.0.13/css/all.css" ↩
         integrity="sha384-DNOHZ68U8hZfKXOrtjWvjxusGo9WQnrNx2sqG0tfsghAvtVlRW3tvkXWZh58N9jp" ↩
         crossorigin="anonymous">
53      <!-- JQuery CDN -->
54  </head>
55  <body>
56  <div class="border ">
57      <div class=".col text-right mr-3 mb-0"> <?php echo $navIdOrButton;?>
58
59          <div id="body" class=" mx-5">
60              <nav class="navbar navbar-expand-lg navbar-light p-0">
61                  <a class="navbar-brand" href="../view/homePage.php" id="logo">
62                      <img src="../public/images/logoSimpleTabGrand.png" alt="logo" width="215" height="125">
63                  </a>
64
65                  <div class="collapse navbar-collapse" id="navbarSupportedContent">
66                      <ul class="navbar-nav mr-auto " style="margin: auto;">
67                          <li class="nav-item active">
68                              <?php echo $navMenu1?>
69                          </li>
70                          <li class="nav-item active">
71                              <?php echo $navMenu2?>
72                          </li>
73                      </ul>
74
75                      <form class="form-inline my-2 my-lg-0" method="get" action="../view/search.php">
76
77                          <input class="form-control mr-sm-0" type="search" placeholder="Rechercher" ↩
     aria-label="Search" id="searchBar" name="nameArtistOrTitleTab">
78                          <button type="submit" class=" btn btn-outline-secondary ">
79                              <i class="fas fa-search"></i>
80                          </button>
81
82                      </form>
83                  </div>
84              </nav>
85          </div>
86      </div>
87  </div>
88
89  </div>
90  <div class=" m-5">
91      <table class="table table-dark">
92          <thead style="color:red">
93          <tr>
94              <th scope="col">Artiste</th>
95              <th scope="col">Titre</th>
96              <th scope="col">Difficulté</th>
97              <th scope="col">Note</th>
98          </tr>
99          </thead>
100         <tbody id='tabs'>
101         </tbody>
102     </table>
103 </div>
104 </body>
105 <script src="../public/js/function.js"></script>
106 <script type="text/javascript">
107     $( document ).ready(function() {
108
109         get_data("../controller/getTabAndRelatedArtist.php",getTabAndRelatedArtist,{},true);
110         function getTabAndRelatedArtist(data) {
111             $('#tabs').empty();
112             data.forEach(function(tablature){
113                 var artist = $('<a class="artistName" >'+ tablature.nameArtist + '</a>');
114                 var td = $("<td>").append(artist);
115                 var lvl = getDifficultyInLetters(tablature.lvlTab);
116                 var tr = $("<tr>").append(td);
117                 $(tr).append(
118                     '<td><a class="titleTab">'+tablature.titleTab+'</a></td>' +
119                     '<td>'+ lvl +'</td>' +
120                     '<td>'+tablature.rateTab+'</td>' +
121                     '</tr>');
122                 $('#tabs').append(tr);
123
124
125                 $(artist).click(function () {
126                     var artistName = $(this).text();
127                     ↩
     get_data("../controller/getTabByArtist.php",getTabByArtist,{'artistName':artistName},true);
128                     function getTabByArtist(data) {
129                         $('#tabs').empty();
130                         data.forEach(function(tablature){
131                             var $lvl = getDifficultyInLetters(tablature.lvlTab);
132                             var $row = $('<tr>' +
133                                 '<td><a class="artistName" >'+ tablature.nameArtist+ '</a></td>' +
```

```
                                '<td>'+tablature.titleTab+'</td>' +
                                '<td>'+ $lvl +'</td>' +
                                '<td>'+tablature.rateTab+'</td>' +
                                '</tr>');
                            $('#tabs').append($row);

                        });
                    }
                });

                $('.titleTab').click(function () {
                    var titleTab = $(this).text();
                    get_data("../controller/getTabByTitle.php",getTabByTitle,{'titleTab':titleTab},true);
                    function getTabByTitle(data) {
                        data.forEach(function(tablature) {
                            window.location.href= "../view/tablaturePage.php?idTab="+tablature.idTab;
                        });
                    }
                });


            });


        }



        function hideModal() {
            $('#addUser').modal('hide')
        }

        $('#btnAddUser').click(function () {
            var name = $('#name').val();
            var forename = $('#forename').val();
            var password = $('#password').val();
            var passwordConfirm = $('#passwordConfirm').val();
            var email = $('#email').val();
            var pseudo = $('#pseudo').val();
            get_data("../controller/getUserByPseudo.php",getUserByPseudo,{'pseudo' : pseudo},true);
            function getUserByPseudo(dataPseudo)
            {
                if(dataPseudo == false)
                {
                    get_data("../controller/getUserByEmail.php",getUserByEmail,{'email' : email},true);
                    function getUserByEmail(dataEmail)
                    {
                        if(dataEmail == false)
                        {
                            get_data("../controller/addUser.php",addUser,{'name' :name, 'forename' : ↩
    forename, 'password' : password,  'email' : email, 'pseudo' : pseudo, 'passwordConfirm' : ↩
    passwordConfirm},true);
                            function addUser(){
                                identifyUser(pseudo,password);
                                $('#addUser').modal('hide');

                            }
                        }
                        else
                        {
                            alert('Ce mail est déjà  pris, veuillez en utiliser un autre');
                        }
                    }

                    }

                else
                {
                    alert('Ce pseudo est déjà  pris, veuillez en utiliser un autre');
                }
            }
        });

        $('#btnIdentifyUser').click(function () {
            var mailOrPseudo = $('#pseudoOrMail').val();
            var pwdConnexion = $('#pwdConnexion').val();
            identifyUser(mailOrPseudo,pwdConnexion);
            $('#connectUser').modal('hide');

        });
    });
</script>
```

1.6.2 search.php

```
<?php
/**
 * Auteur: romain.ssr@eduge.ch
 * Date : 17.05.2018
 * projet: SimpleTab
```

```php
 * description : page de résultat des recherches
 */


session_start();

require_once"modals.php";

$navIdOrButton="";
$navMenu1 = "";
$navMenu2 = "";

if(isset($_SESSION['user']))
{
    $navIdOrButton = "<div class=\"dropdown\">
                            <button type=\"button\" onclick=\" \" class=\"btn btn-light dropdown-toggle\" data-toggle=\"dropdown\" aria-haspopup=\"true\" aria-expanded=\"false\"> ".$_SESSION['user'][0]['pseudoUser']."</button>
                            <div class=\"dropdown-menu\" aria-labelledby=\"dropdownMenuButton\">
                                <a class=\"dropdown-item\" href=\"../controller/destroySession.php\">Déconnexion</a>
                            </div>
                    </div>";
    if($_SESSION['user'][0]['role_idrole'] == 0)
    {
        $navMenu1 = "<h5><a class=\"nav-link\" href=\"../view/homePage.php\">Accueil <span class=\"sr-only\">(current)</span></a></h5>";
        $navMenu2 = "<h5><a class=\"nav-link\" href=\"../view/tablatureManagerPage.php\">Gestion des tablatures </a></h5>";
    }
    elseif($_SESSION['user'][0]['role_idrole'] == 1)
    {
        $navMenu1 ="<h5><a class=\"nav-link\" href=\"../view/homePage.php\">Accueil <span class=\"sr-only\">(current)</span></a></h5>";
        $navMenu2 = "<h5><a class=\"nav-link\" href=\"../view/tablatureAndUserManagerPage.php\">Gestion des tablatures et utilisateurs </a></h5>";
    }
}
else
{
    $navIdOrButton = "<button type=\"button\" class=\"btn btn-light\" data-toggle=\"modal\" data-target=\"#addUser\">S'inscrire</button> | <button type=\"button\" class=\"btn btn-light\"data-toggle=\"modal\" data-target=\"#connectUser\">S'identifier</button>";
    $navMenu1 = "<h5><a class=\"nav-link\" href=\"../view/homePage.php\">Accueil <span class=\"sr-only\">(current)</span></a></h5>";
}
?>
<! DOCTYPE HTML>
<html>
<head>
    <title>Recherche</title>
    <script src="https://code.jquery.com/jquery-3.3.1.js" integrity="sha256-2Kok7MbOyxpgUVvAk/HJ2jigOSYS2auK4Pfzbm7uH60=" crossorigin="anonymous"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js" integrity="sha384-ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/l8WvCWPIPm49" crossorigin="anonymous"></script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.1/js/bootstrap.min.js" integrity="sha384-smHYKdLADwkXOn1EmN1qk/HfnUcbVRZyYmZ4qpPea6sjB/pTJ0euyQp0Mk8ck+5T" crossorigin="anonymous"></script>
    <link rel="stylesheet" type="text/css" href="../public/css/style.css">
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.0/css/bootstrap.min.css" integrity="sha384-9gVQ4dYFwwWSjIDZnLEWnxCjeSWFphJiwGPXr1jddIhOegiu1FwO5qRGvFXOdJZ4" crossorigin="anonymous">

    <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.0.13/css/all.css" integrity="sha384-DNOHZ68U8hZfKXOrtjWvjxusGo9WQnrNx2sqG0tfsghAvtVlRW3tvkXWZh58N9jp" crossorigin="anonymous">
    <!-- JQuery CDN -->
</head>
<body>
<div class="border ">
    <div class=".col text-right mr-3 mb-0"> <?php echo $navIdOrButton;?>

        <div id="body" class=" mx-5">
            <nav class="navbar navbar-expand-lg navbar-light p-0">
                <a class="navbar-brand" href="../view/homePage.php" id="logo">
                    <img src="../public/images/logoSimpleTabGrand.png" alt="logo" width="215" height="125">
                </a>

                <div class="collapse navbar-collapse" id="navbarSupportedContent">
                    <ul class="navbar-nav mr-auto " style="margin: auto;">
                        <li class="nav-item active">
                            <?php echo $navMenu1?>
                        </li>
                        <li class="nav-item active">
                            <?php echo $navMenu2?>
                        </li>
                    </ul>

                    <form class="form-inline my-2 my-lg-0">

                        <input class="form-control mr-sm-0" type="search" placeholder="Rechercher" aria-label="Search" id="searchBar">
```

```
                        <button type="button" class=" btn btn-outline-secondary " onclick="search($('#searchBar').val())">
                            <i class="fas fa-search"></i>
                        </button>

                    </form>
                </div>
            </nav>
        </div>
    </div>
</div>

</div>
<div class=" m-5">
    <table class="table table-dark">
        <thead style="color:red">
        <tr>
            <th scope="col">Artiste</th>
            <th scope="col">Titre</th>
            <th scope="col">Difficulté</th>
            <th scope="col">Note</th>
        </tr>
        </thead>
        <tbody id='tabs'>
        </tbody>
    </table>
</div>
</body>
<script src="../public/js/function.js"></script>
<script type="text/javascript">
    $( document ).ready(function() {
            var tabTitleOrArtistNAme = <?php echo '"'; echo $_GET['nameArtistOrTitleTab']; echo '"';?>;
            get_data("../controller/getTabAndRelatedArtistByName.php",getTabAndRelatedArtistByName,{'tabTitleOrArtistNAme' :tabTitleOrArtistNAme},true);
            function getTabAndRelatedArtistByName(data)
            {
                $('#tabs').empty();
                data.forEach(function(tablature){
                    var artist = $('<a class="artistName" >'+ tablature.nameArtist + '</a>');
                    var td = $("<td>").append(artist);
                    var lvl = getDifficultyInLetters(tablature.lvlTab);
                    var tr = $("<tr>").append(td);
                    $(tr).append(
                        '<td>'+tablature.titleTab+'</td>' +
                        '<td>'+ lvl +'</td>' +
                        '<td>'+tablature.rateTab+'</td>' +
                        '</tr>');
                    $('#tabs').append(tr);


                    $(artist).click(function () {
                        var artistName = $(this).text();
                        get_data("../controller/getTabByArtist.php",getTabByArtist,{'artistName':artistName},true);
                        function getTabByArtist(data) {
                            $('#tabs').empty();
                            data.forEach(function(tablature){
                                var $lvl = getDifficultyInLetters(tablature.lvlTab);
                                var $row = $('<tr>' +
                                    '<td><a class="artistName" >'+ tablature.nameArtist+ '</a></td>' +
                                    '<td>'+tablature.titleTab+'</td>' +
                                    '<td>'+ $lvl +'</td>' +
                                    '<td>'+tablature.rateTab+'</td>' +
                                    '</tr>');
                                $('#tabs').append($row);

                            });
                        }
                    });
                });
            }

        $('#btnAddUser').click(function () {
            var name = $('#name').val();
            var forename = $('#forename').val();
            var password = $('#password').val();
            var passwordConfirm = $('#passwordConfirm').val();
            var email = $('#email').val();
            var pseudo = $('#pseudo').val();
            get_data("../controller/getUserByPseudo.php",getUserByPseudo,{'pseudo' : pseudo},true);
            function getUserByPseudo(dataPseudo)
            {
                if(dataPseudo == false)
                {
                    get_data("../controller/getUserByEmail.php",getUserByEmail,{'email' : email},true);
                    function getUserByEmail(dataEmail)
                    {
                        if(dataEmail == false)
                        {
                            get_data("../controller/addUser.php",addUser,{'name' :name, 'forename' : forename, 'password' : password,  'email' : email, 'pseudo' : pseudo, 'passwordConfirm' : passwordConfirm},true);
                            function addUser(){
```

```
                                identifyUser(pseudo,password);
                                $('#addUser').modal('hide');

                            }
                        }
                        else
                        {
                            alert('Ce mail est déjà  pris, veuillez en utiliser un autre');
                        }
                    }

                }

                else
                {
                    alert('Ce pseudo est déjà  pris, veuillez en utiliser un autre');
                }
            }
        });

        $('#btnIdentifyUser').click(function () {
            var mailOrPseudo = $('#pseudoOrMail').val();
            var pwdConnexion = $('#pwdConnexion').val();
            identifyUser(mailOrPseudo,pwdConnexion);
            $('#connectUser').modal('hide');

        });
    });
</script>
```

1.6.3 tablaturePage.php

```
<?php
/**
 * Auteur: romain.ssr@eduge.ch
 * Date : 17.05.2018
 * projet: SimpleTab
 * description : page des détails d'une tablature
 */

session_start();

require_once "../view/modals.php";

if(isset($_SESSION['user']))
{
  $userId =  $_SESSION['user'][0]['idUsers'];
  $pseudoUSer = $_SESSION['user'][0]['pseudoUser'];
}

if(isset($_GET['idTab']))
{
    $idTab = $_GET['idTab'];
    include "../tabs/".$idTab.".php";
    $tabs = new SimpleXMLElement($xmlstr);

    $idTab = $_GET['idTab'];
    $title =$tabs->metadata->title;
    $nameArtist =$tabs->metadata->author;
    $tuning = $tabs->metadata->tuning;
    $capo =$tabs->metadata->capo;
    $key =$tabs->metadata->key;
    $lvl =$tabs->metadata->level;
    $bodyTab = $tabs->corpse;

    $body = "<div id='tab' class='p-3 border' style='display: table; margin:0 auto;'>
    <div id='metadatas' >
        <table>
            <tr>
                <td> Titre :</td>
                <td>$title </td>
            </tr>
            <tr>
                <td> Auteur :</td>
                <td> $nameArtist</td>
            </tr>
            <tr>
                <td> Accordage :</td>
                <td> $tuning</td>
            </tr>
            <tr>
                <td> Capo :</td>
                <td>$capo </td>
            </tr>
            <tr>
                <td> Tonalité :</td>
                <td> $key</td>
            </tr>
            <tr>
                <td> Difficulté :</td>
```

```
                <td>$lvl </td>
            </tr>
        </table>
    </div>
    <div id='tabBody' >
        <table>
            <tr>
                <td>
                    <pre>$bodyTab </pre>
                </td>

            </tr>
        </table>
    </div>
   <div class='border-top' id='commentSection'>
    <table class='col'>
   <tr>
            <td colspan='2' id='Score' class='text-right'><i id='1' class='far fa-star rateStar'></i><i id='2' class='far fa-star rateStar'></i><i id='3' class='far fa-star rateStar'></i><i id='4' class='far fa-star rateStar'></i><i id='5' class='far fa-star rateStar'></i></td>
        </tr>
            <td colspan='2' id=\"averageScore\" class='text-right'> note moyenne : </td>
        </tr>
        <tr id='comments'>
            <td colspan='2'>Commentaire :</td>
        </tr>
        <tr id='addComment'>
            <td colspan='2'><textarea style='width: 100%;' id='myComment' placeholder='Que pensez-vous de cette tablature ?'></textarea><button style='float: right; display: inline;' id='postComment'>Soumettre</button></td>
        </tr>
    </table>
</div>
</div>

";




}
else
{
    $body = "<div class=\"alert alert-danger text-center\" role=\"alert\">
                Aucune Tablature sélectionée, veuillez retourner à  <a class=\"alert-link\" href='../view/homePage.php '>la page d'Accueil</a> et cliquez sur le titre pour sélectionner une tablature.
            </div>";
}

    $navIdOrButton="";
    $navMenu1 = "";
    $navMenu2 = "";

if(isset($_SESSION['user']))
{
    $navIdOrButton = "<div class=\"dropdown\">
                            <button type=\"button\" onclick=\" \" class=\"btn btn-light dropdown-toggle\" data-toggle=\"dropdown\" aria-haspopup=\"true\" aria-expanded=\"false\"> ".$_SESSION['user'][0]['pseudoUser']."</button>
                            <div class=\"dropdown-menu\" aria-labelledby=\"dropdownMenuButton\">
                                <a class=\"dropdown-item\" href=\"../controller/destroySession.php\">Déconnexion</a>
                            </div>
                        </div>";
    if($_SESSION['user'][0]['role_idrole'] == 0)
    {
        $navMenu1 = "<h5><a class=\"nav-link\" href=\"../view/homePage.php\">Accueil <span class=\"sr-only\">(current)</span></a></h5>";
        $navMenu2 = "<h5><a class=\"nav-link\" href=\"../view/tablatureManagerPage.php\">Gestion des tablatures </a></h5>";
    }
    elseif($_SESSION['user'][0]['role_idrole'] == 1)
    {
        $navMenu1 ="<h5><a class=\"nav-link\" href=\"../view/homePage.php\">Accueil <span class=\"sr-only\">(current)</span></a></h5>";
        $navMenu2 = "<h5><a class=\"nav-link\" href=\"../view/tablatureAndUserManagerPage.php\">Gestion des tablatures et utilisateurs </a></h5>";
    }
}
else
{
    $navIdOrButton = "<button type=\"button\" class=\"btn btn-light\" data-toggle=\"modal\" data-target=\"#addUser\">S'inscrire</button> | <button type=\"button\" class=\"btn btn-light\"data-toggle=\"modal\" data-target=\"#connectUser\">S'identifier</button>";
    $navMenu1 = "<h5><a class=\"nav-link\" href=\"../view/homePage.php\">Accueil <span class=\"sr-only\">(current)</span></a></h5>";
}

?>

<! DOCTYPE HTML>
<html>
<head>
```

```
    <title>Page d'une tablature</title>
    <script src="https://code.jquery.com/jquery-3.3.1.js" ↩
     integrity="sha256-2Kok7MbOyxpgUVvAk/HJ2jigOSYS2auK4Pfzbm7uH60=" crossorigin="anonymous"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js" ↩
     integrity="sha384-ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/l8WvCWPIPm49" ↩
     crossorigin="anonymous"></script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.1/js/bootstrap.min.js" ↩
     integrity="sha384-smHYKdLADwkXOn1EmN1qk/HfnUcbVRZyYmZ4qpPea6sjB/pTJ0euyQp0Mk8ck+5T" ↩
     crossorigin="anonymous"></script>
    <link rel="stylesheet" type="text/css" href="../public/css/style.css">
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.0/css/bootstrap.min.css" ↩
     integrity="sha384-9gVQ4dYFwwWSjIDZnLEWnxCjeSWFphJiwGPXr1jddIhOegiu1FwO5qRGvFXOdJZ4" ↩
     crossorigin="anonymous">

    <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.0.13/css/all.css" ↩
     integrity="sha384-DNOHZ68U8hZfKXOrtjWvjxusGo9WQnrNx2sqG0tfsghAvtVlRW3tvkXWZh58N9jp" ↩
     crossorigin="anonymous">
    <!-- JQuery CDN -->
</head>
<body>
<div class="border ">
    <div class=".col text-right mr-3 mb-0"> <?php echo $navIdOrButton;?>

        <div id="body" class=" mx-5">
            <nav class="navbar navbar-expand-lg navbar-light  p-0">
                <a class="navbar-brand" href="../view/homePage.php" id="logo">
                    <img src="../public/images/logoSimpleTabGrand.png" alt="logo" width="215" height="125">
                </a>

                <div class="collapse navbar-collapse" id="navbarSupportedContent">
                    <ul class="navbar-nav mr-auto " style="margin: auto;">
                        <li class="nav-item active">
                            <?php echo $navMenu1?>
                        </li>
                        <li class="nav-item active">
                            <?php echo $navMenu2?>
                        </li>
                    </ul>

                    <form class="form-inline my-2 my-lg-0" method="get" action="../view/search.php">

                        <input class="form-control mr-sm-0" type="search" placeholder="Rechercher" ↩
     aria-label="Search" id="searchBar" name="nameArtistOrTitleTab">
                        <button type="submit" class=" btn btn-outline-secondary ">
                            <i class="fas fa-search"></i>
                        </button>

                    </form>
                </div>
            </nav>
        </div>
    </div>
</div>
<?php echo $body; ?>
<script src="../public/js/function.js"></script>
<script type="text/javascript">
    $( document ).ready(function() {
        var idTab = <?php echo $_GET['idTab'];?>;
        var idUserComment = <?php if(isset($userId)){echo $userId;}else{echo -1;} ?>;
        var idTabComment = <?php  if(isset($idTab)){echo $idTab;}else{echo -1;}?>;
        var pseudoUser = "<?php if(isset($pseudoUSer)){echo $pseudoUSer;}else{echo "";} ?>";
        get_data("../controller/getCommentsByTab.php",getCommentsByTab,{'idTab': idTab},true);
        function getCommentsByTab(data) {
            data.forEach(function (comment) {
                $('#comments').after('<tr class="border-top p-1 m-1"><td ><label><b>'+comment.pseudoUser+': ↩
     </b></label><label> '+comment.contentComment+'</label></td></tr>')
            });
        }

        get_data("../controller/getRateByTabId.php",getRateByTabId,{'idTab': idTab},true);
        function getRateByTabId(data)
        {
            data.forEach(function (rate) {
                if (rate.users_idUsers == idUserComment) {
                    $('#Score').text("Vous avez attribué la note de " + rate.rate );
                }
            });
        }

        get_data("../controller/getTabById.php",getTabById,{'idTab': idTab},true);
        function getTabById(data) {
            data.forEach (function (tab) {
                if(tab.rateTab > 0)
                {
                    $('#averageScore').text("note moyenne : "+tab.rateTab);
                }
                else
                {
                    $('#averageScore').text("La tablature n'est pas notée");
                }
            });


        }
```

```
220
221
222
223         $('#postComment').click(function () {
224
225
226             var contentComment = $('#myComment').val();
227
228
229             if(idUserComment != -1 && idTabComment != -1)
230             {
231             get_data("../controller/addComment.php", addComment, {
232                 'contentComment': contentComment,
233                 'idTabComment': idTabComment,
234                 'idUserComment': idUserComment
235             }, true);
236
237             function addComment(data) {
238
239                 if (data == true) {
240                     alert("votre commentaire a bien été enregistré !");
241                     location.reload();
242                 }
243                 else {
244                     alert("un problème est survenu");
245                 }
246             }
247             }
248             else
249             {
250                 alert("vous devez être connecté pour laisser une appréciation");
251             }
252         });
253
254         $('#btnAddUser').click(function () {
255             var name = $('#name').val();
256             var forename = $('#forename').val();
257             var password = $('#password').val();
258             var passwordConfirm = $('#passwordConfirm').val();
259             var email = $('#email').val();
260             var pseudo = $('#pseudo').val();
261             get_data("../controller/getUserByPseudo.php",getUserByPseudo,{'pseudo' : pseudo},true);
262             function getUserByPseudo(dataPseudo)
263             {
264                 if(dataPseudo == false)
265                 {
266                     get_data("../controller/getUserByEmail.php",getUserByEmail,{'email' : email},true);
267                     function getUserByEmail(dataEmail)
268                     {
269                         if(dataEmail == false)
270                         {
271                             get_data("../controller/addUser.php",addUser,{'name' :name, 'forename' : ↩
    forename, 'password' : password,  'email' : email, 'pseudo' : pseudo, 'passwordConfirm' : ↩
    passwordConfirm},true);
272                             function addUser(){
273                                 identifyUser(pseudo,password);
274                                 $('#addUser').modal('hide');
275
276                             }
277                         }
278                         else
279                         {
280                             alert('Ce mail est déjà  pris, veuillez en utiliser un autre');
281                         }
282                     }
283
284                 }
285
286                 else
287                 {
288                     alert('Ce pseudo est déjà  pris, veuillez en utiliser un autre');
289                 }
290             }
291         });
292
293
294         $('#btnIdentifyUser').click(function () {
295             var mailOrPseudo = $('#pseudoOrMail').val();
296             var pwdConnexion = $('#pwdConnexion').val();
297             identifyUser(mailOrPseudo,pwdConnexion);
298             $('#connectUser').modal('hide');
299
300         });
301
302         $('.rateStar').mouseenter(function () {
303             for(i = $(this).attr('id');i>=1;i--)
304             {
305                 $('#'+i).removeClass("far fa-star");
306                 $('#'+i).addClass("fas fa-star");
307             }
308
309
310         });
311         $('.rateStar').mouseleave(function () {
```

```
            $(this).removeClass("fas fa-star");
            $(this).addClass("far fa-star");
        });

        $('.rateStar').click(function () {

            if(idUserComment != -1) {
                var rate = $(this).attr('id');

                get_data("../controller/addRate.php", addRate, {
                    'idUser': idUserComment,
                    'idTab': idTab,
                    'rate': rate
                }, true);

                function addRate(data) {
                    if (data == true && pseudoUser != "") {

                        alert("votre note a bien été enregistrée !");
                        location.reload();
                    }
                    else {
                        alert("un problème est survenu");
                    }
                }
            }
            else
            {
                alert("vous devez être connecté pour laisser une appréciation");
            }
        });

    });
</script>
```

1.6.4 tablatureManagerPage.php

```
<?php
/**
 * Auteur: romain.ssr@eduge.ch
 * Date : 16.05.2018
 * projet: SimpleTab
 * description : page de gestion de l'utilisateur
 */

session_start();

require_once "../view/modals.php";

$navIdOrButton="";
$navMenu1 = "";
$navMenu2 = "";
$redirect = 1;

if(isset($_SESSION['user']))
{
    $navIdOrButton = "<div class=\"dropdown\">
                            <button type=\"button\" onclick=\" \" class=\"btn btn-light dropdown-toggle\" data-toggle=\"dropdown\" aria-haspopup=\"true\" aria-expanded=\"false\">
                            ".$_SESSION['user'][0]['pseudoUser']."</button>
                            <div class=\"dropdown-menu\" aria-labelledby=\"dropdownMenuButton\">
                                <a class=\"dropdown-item\" href=\"../controller/destroySession.php\">Déconnexion</a>
                            </div>
                    </div>";
    if($_SESSION['user'][0]['role_idrole'] == 0)
    {
        $navMenu1 = "<h5><a class=\"nav-link\" href=\"../view/homePage.php\">Accueil <span class=\"sr-only\">(current)</span></a></h5>";
        $navMenu2 = "<h5><a class=\"nav-link\" href=\"../view/tablatureManagerPage.php\">Gestion des tablatures </a></h5>";
    }
    elseif($_SESSION['user'][0]['role_idrole'] == 1)
    {
        $navMenu1 ="<h5><a class=\"nav-link\" href=\"../view/homePage.php\">Accueil <span class=\"sr-only\">(current)</span></a></h5>";
        $navMenu2 = "<h5><a class=\"nav-link\" href=\"../view/tablatureAndUserManagerPage.php\">Gestion des tablatures et utilisateurs </a></h5>";
    }
}
else
{
   $redirect = 0;
}
?>
<html>
<head>
    <title>Gestion des tablatures</title>
    <script src="https://code.jquery.com/jquery-3.3.1.js"
     integrity="sha256-2Kok7MbOyxpgUVvAk/HJ2jigOSYS2auK4Pfzbm7uH60=" crossorigin="anonymous"></script>
```

```
46      <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js" ↩
         integrity="sha384-ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/l8WvCWPIPm49" ↩
         crossorigin="anonymous"></script>
47      <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.1/js/bootstrap.min.js" ↩
         integrity="sha384-smHYKdLADwkXOn1EmN1qk/HfnUcbVRZyYmZ4qpPea6sjB/pTJ0euyQp0Mk8ck+5T" ↩
         crossorigin="anonymous"></script>
48      <!-- Copyright (c) 2016 Indri Muska. Licensed under the MIT license.-->
49      <script ↩
         src="//rawgithub.com/indrimuska/jquery-editable-select/master/dist/jquery-editable-select.min.js"></script>
50      <link ↩
         href="//rawgithub.com/indrimuska/jquery-editable-select/master/dist/jquery-editable-select.min.css" ↩
         rel="stylesheet">
51
52      <link rel="stylesheet" type="text/css" href="../public/css/style.css">
53      <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.0/css/bootstrap.min.css" ↩
         integrity="sha384-9gVQ4dYFwwWSjIDZnLEWnxCjeSWFphJiwGPXr1jddIhOegiu1FwO5qRGvFXOdJZ4" ↩
         crossorigin="anonymous">
54
55      <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.0.13/css/all.css" ↩
         integrity="sha384-DNOHZ68U8hZfKXOrtjWvjxusGo9WQnrNx2sqG0tfsghAvtVlRW3tvkXWZh58N9jp" ↩
         crossorigin="anonymous">
56      <!-- JQuery CDN -->
57  </head>
58  <body>
59  <div class="border ">
60      <div class=".col text-right mr-3 mb-0"> <?php echo $navIdOrButton;?>
61
62          <div id="body" class=" mx-5">
63              <nav class="navbar navbar-expand-lg navbar-light  p-0">
64                  <a class="navbar-brand" href="../view/homePage.php" id="logo">
65                      <img src="../public/images/logoSimpleTabGrand.png" alt="logo" width="215" height="125">
66                  </a>
67
68                  <div class="collapse navbar-collapse" id="navbarSupportedContent">
69                      <ul class="navbar-nav mr-auto " style="margin: auto;">
70                          <li class="nav-item active">
71                              <?php echo $navMenu1?>
72                          </li>
73                          <li class="nav-item active">
74                              <?php echo $navMenu2?>
75                          </li>
76                      </ul>
77                      <form class="form-inline my-2 my-lg-0" method="get" action="../view/search.php">
78                          <input class="form-control mr-sm-0" type="search" placeholder="Rechercher" ↩
     aria-label="Search" id="searchBar" name="nameArtistOrTitleTab">
79                          <button type="submit" class=" btn btn-outline-secondary ">
80                              <i class="fas fa-search"></i>
81                          </button>
82                      </form>
83                  </div>
84              </nav>
85          </div>
86      </div>
87  </div>
88  <div  id="message" style="display:none;">
89  </div>
90
91  <div class=" m-5" >
92
93      <table class="table table-dark">
94          <thead style="color:red">
95          <tr class="text-center">
96              <th colspan="6"><button class="btn btn-dark" data-toggle="modal" data-target="#addTab" ↩
     style="color:red;">Ajouter une tablature</button></a></i></th>
97          </tr>
98          <tr>
99              <th scope="col">Artiste</th>
100             <th scope="col">Titre</th>
101             <th scope="col">Difficulté</th>
102             <th scope="col">Note</th>
103             <th scope="col" colspan="2" style="text-align: center">Gestion</th>
104         </tr>
105         </thead>
106         <tbody id='tabs'>
107         </tbody>
108     </table>
109     </td>
110
111
112 </div>
113 </body>
114 <script src="../public/js/function.js"></script>
115 <script type="text/javascript">
116     $( document ).ready(function() {
117
118         var redirect = <?php echo $redirect;?>;
119
120         if(redirect == 1 ) {
121
122
123             var idUser = <?php if(isset($_SESSION['user'][0]['idUsers'])){ echo ↩
     $_SESSION['user'][0]['idUsers'];} else {echo -1;}?>;
124             reloadTabByUSers(idUser);
125
```

```
126             get_data("../controller/getArtistNames.php",getArtistNames,{},true);
127             function getArtistNames(data)
128             {
129                 data.forEach(function (artist) {
130                     $('#addArtist').append('<option>'+artist.nameArtist+'</option>');
131                     $('#modifyAuthor').append('<option>'+artist.nameArtist+'</option>');
132                 })
133 
134             }
135             $('#addArtist').editableSelect({ effects: 'default' });
136             $('#modifyAuthor').editableSelect({ effects: 'default' });
137 
138 
139             $('#btnAddTab').click(function () {
140                 var title = $('#addTitle').val();
141                 var artist = $('#addArtist').val();
142                 var lvl = $('#addLvl option:selected').val();
143                 var capo = $('#addCapo').val();
144                 var key = $('#addKey').val();
145                 var tuning = $('#addTuning').val();
146                 var tabBody = $('#addTablatureBody').val();
147                 get_data("../controller/addTab.php", addTab, {
148                     'addTitle': title,
149                     'addArtist': artist,
150                     'addLvl': lvl,
151                     'addCapo': capo,
152                     'addKey': key,
153                     'addTuning': tuning,
154                     'addTabBody': tabBody,
155                 }, false);
156 
157                 function addTab(data) {
158                     var message = "";
159                     if (data == true) {
160                         alert("La tablature a bien été ajoutée, elle sera visible quand l'administrateur ↩
    l'aura acceptée")
161                     }
162                     else {
163                         alert("Un problème est survenu")
164 
165                     }
166                     reloadTabByUSers(idUser);
167 
168 
169                 }
170 
171                 $('#addTab').modal('hide')
172                 reloadTabByUSers(idUser);
173             });
174 
175 
176             $('#btnModifyTab').click(function () {
177                 var title = $('#modifyTitle').val();
178                 var artist = $('#modifyAuthor').val();
179                 var lvl = $('#modifyLvl option:selected').val();
180                 var capo = $('#modifyCapo').val();
181                 var key = $('#modifyKey').val();
182                 var tuning = $('#modifyTuning').val();
183                 var tabBody = $('#modifyTablatureBody').val();
184                 get_data("../controller/modifyTabById.php", modifyTabById, {
185                     'modifyTitle': title,
186                     'modifyAuthor': artist,
187                     'modifyLvl': lvl,
188                     'modifyCapo': capo,
189                     'modifyKey': key,
190                     'modifyTuning': tuning,
191                     'modifyTabBody': tabBody,
192 
193                 }, false);
194 
195                 function modifyTabById(data) {
196                     if (data == true) {
197                         alert('La tablature a été modifiée avec succès');
198                     }
199                     else {
200                         alert("Un problème est survenu")
201                     }
202                     reloadTabByUSers(idUser);
203 
204 
205                 }
206 
207                 $('#modifyTab').modal('hide')
208                 reloadTabByUSers(idUser);
209             });
210         }
211         else
212         {
213             window.location.href = "../view/homePage.php";
214         }
215     });
216 </script>
```

1.6.5 modals.php

```php
<?php
/**
 * Auteur: romain.ssr@eduge.ch
 * Date : 11.05.2018
 * projet: SimpleTab
 * description : modals du site
 */

?>
<!-- Modal Pop-up addUser() -->
<div class="modal fade" id="addUser" role="dialog">
    <div class="modal-dialog modal-dialog-centered">

        <!-- Modal content-->
        <div class="modal-content ">
            <div class="modal-header pb-0">
                <table>
                    <tr>
                        <td>
                            <h4 class="modal-title " style="text-align: left;">Inscription</h4>
                        </td>
                    </tr>
                    <tr>
                        <td> <small class="text-muted"> Les champs marqués d'un <label style="color: red;">*</label> sont obligatoires</small></td>
                    </tr>
            </table>


            </div>

            <div class="modal-body">
                <table class="col">
                    <tr>
                        <td> <label class="text-left">Nom  <label style="color: red;">*</label> : </label></td>
                        <td class="float-right"> <input type="text"  class=" emptyForbiddenField" id="name" style="resize: none;" autofocus ></td>
                    </tr>
                    <tr>
                        <td><label class="text-left">Prénom <label style="color: red;">*</label> : </label></td>
                        <td class="float-right"><input type="text"  class=" emptyForbiddenField" id="forename" style="resize: none;" ></td>
                    </tr>
                    <tr>
                        <td><label class="text-left">Pseudo<label style="color: red;">*</label> : </label></td>
                        <td class="float-right"><input type="text"  class=" emptyForbiddenField" id="pseudo" style="resize: none;" ></td>
                    </tr>
                    <tr>
                        <td><label class="text-left">Email <label style="color: red;">*</label> : </label></td>
                        <td class="float-right"><input type="email"  class=" emptyForbiddenField" id="email" style="resize: none;" ></td>
                    </tr>
                    <tr>
                        <td><label class="text-left">Mot de passe <label style="color: red;">*</label> : </label></td>
                        <td class="float-right"> <input type="password"  class=" emptyForbiddenField" id="password" style="resize: none;" ></td>
                    </tr>
                    <tr>
                        <td><label class="text-left">Retapez le mot de passe <label style="color: red;">*</label> : </label></td>
                        <td class="float-right"><input type="password"  class=" emptyForbiddenField" id="passwordConfirm" style="resize: none;" ></td>
                    </tr>
                    <tr>
                        <td colspan="2"><button style="color:red;" id="btnAddUser" type="button" class="btn btn-dark col">Rejoindre SimpleTab</button></td>
                    </tr>
                </table>
            </div>
            <!-- Modal footer-->
            <div class="modal-footer">
                <table class="col text-center">
                    <tr>
                        <td><label> Déjà  inscrit ? </label></td>
                    </tr>
                    <tr>
                        <td>
                            <button type="button" class="btn btn-light" data-toggle="modal" data-target="#connectUser" onclick="hideModal()">Identifiez-vous !</button>
                        </td>
                    </tr>
                </table>
            </div>
        </div>
    </div>
```

```
</div>
<!-- END Modal -->


<!-- Modal Pop-up connectUser() -->
<div class="modal fade" id="connectUser" role="dialog">
    <div class="modal-dialog modal-dialog-centered">

        <!-- Modal content-->
        <div class="modal-content">
            <div class="modal-header pb-0">

            <h4 class="modal-title " style="text-align: left;">Connexion</h4>
            </div>
            <div class="modal-body">
                <table class="col">
                    <tr>
                        <td> <label class="text-left">Pseudo ou email: </label></td>
                        <td class="float-right"> <input type="text"  class="mb-2  emptyForbiddenField" id="pseudoOrMail" style="resize: none;" autofocus></td>
                    </tr>
                    <tr>
                        <td><label class="text-left">Mot de passe: </label></td>
                        <td class="float-right"><input type="password"  class=" emptyForbiddenField" id="pwdConnexion" style="resize: none;" ></td>
                    </tr>
                    <tr >
                        <td colspan="2" ><button style="color:red;" id="btnIdentifyUser" type="button" class="btn btn-dark col mt-2">Connexion</button></td>
                    </tr>
                </table>
            </div>

        </div>
    </div>
</div>
<!-- END Modal -->

<!-- Modal Add Tab Form -->
<div class="modal fade " id="addTab" role="dialog">
    <div class="modal-dialog modal-dialog-centered modal-lg" style="display: table;">

        <!-- Modal content-->
        <div class="modal-content">
            <div class="modal-header pb-0">

                <h4 class="modal-title " style="text-align: left;">Ajouter une tablature</h4>
            </div>
            <div class="modal-body">
                <table class="col">
                    <tr>
                        <td><label class="text-left">Titre: </label></td>
                        <td><label class="text-left">Auteur: </label></td>
                        <td><label class="text-left">Difficulté: </label></td>
                        <td><label class="text-left">Capo: </label></td>
                        <td><label class="text-left">Tonalité: </label></td>
                        <td><label class="text-left">Accordage: </label></td>
                    </tr>
                    <tr>
                        <td class="pr-2"> <input type="text"  class=" emptyForbiddenField" id="addTitle" style="resize: none;" autofocus></td>
                        <td class="pr-2"> <select type="text"  class=" emptyForbiddenField" id="addArtist" style="resize: none;" >
                            </select></td>
                        <td class="pr-2" style="display: inline"><select class="form-control" id="addLvl">
                                <option value="0">Facile</option>
                                <option value="1">Moyen</option>
                                <option value="2">Difficile</option>

                            </select></td>
                        <td class="pr-2"> <input type="text"  class=" emptyForbiddenField" id="addCapo" style="resize: none;" ></td>
                        <td class="pr-2"> <input type="text"  class=" emptyForbiddenField" id="addKey" style="resize: none;" ></td>
                        <td class="pr-2"> <input type="text"  class=" emptyForbiddenField" id="addTuning" style="resize: none;" value="E A D G B E"></td>
                    </tr>
                    <tr>
                        <td>Tablature</td>
                    </tr>
                    <tr>
                        <td colspan="6" ><textarea style="resize: vertical; height: 100%; width: 100%;" id="addTablatureBody"></textarea></td>
                    </tr>
                    <tr >

                        <td colspan="6" ><button style="color:red;" id="btnAddTab" type="button" class="btn btn-dark col mt-2">Ajouter</button></td>
                    </tr>
                </table>
            </div>
        </div>
    </div>
</div>
```

```
<!-- END Modal -->

<!-- Modal Modify Tab Form -->
<div class="modal fade bd-example-modal-lg" id="modifyTab" role="dialog">
    <div class="modal-dialog modal-dialog-centered modal-lg" style="display: table;">

        <!-- Modal content-->
        <div class="modal-content">
            <div class="modal-header pb-0">

                <h4 class="modal-title " style="text-align: left;">Modifier une tablature</h4>
            </div>
            <div class="modal-body">
                <table class="col">
                    <tr>
                        <td> <label class="text-left">Titre: </label></td>
                        <td><label class="text-left">Auteur: </label></td>
                        <td><label class="text-left">Difficulté: </label></td>
                        <td><label class="text-left">Capo: </label></td>
                        <td><label class="text-left">Tonalité: </label></td>
                        <td><label class="text-left">Accordage: </label></td>
                    </tr>
                    <tr>
                        <td class="pr-2"> <input type="text"  class=" emptyForbiddenField" id="modifyTitle" style="resize: none;" autofocus></td>
                        <td class="pr-2"> <select type="text"  class=" emptyForbiddenField" id="modifyAuthor" style="resize: none;" >
                            </select></td>
                        <td class="pr-2"><select class="form-control" id="modifyLvl">
                                <option value="0">Facile</option>
                                <option value="1">Moyen</option>
                                <option value="2">Difficile</option>

                            </select></td>
                        <td class="pr-2"> <input type="text"  class=" emptyForbiddenField" id="modifyCapo" style="resize: none;" ></td>
                        <td class="pr-2"> <input type="text"  class=" emptyForbiddenField" id="modifyKey" style="resize: none;" ></td>
                        <td class="pr-2"> <input type="text"  class=" emptyForbiddenField" id="modifyTuning" style="resize: none;" ></td>
                    </tr>
                    <tr>
                        <td>Tablature</td>
                    </tr>
                    <tr>
                        <td colspan="6" ><textarea style="resize: vertical; height: 100%; width: 100%;" id="modifyTablatureBody"></textarea></td>
                    </tr>
                    <tr >

                        <td colspan="6" ><button style="color:red;" id="btnModifyTab" type="button" class="btn btn-dark col mt-2">Modifier</button></td>
                    </tr>
                </table>
            </div>
        </div>
    </div>
</div>
<!-- END Modal -->
```

1.6.6 tablatureAndUserManagerPage.php

```
<?php
/**
 * Auteur: romain.ssr@eduge.ch
 * Date : 22.05.2018
 * projet: SimpleTab
 * description : page de gestion de l'administrateur
 */


session_start();

require_once"modals.php";

$navIdOrButton="";
$navMenu1 = "";
$navMenu2 = "";
$redirect = 1;


if(isset($_SESSION['user']))
{
    $navIdOrButton = "<div class=\"dropdown\">
                        <button type=\"button\" onclick=\" \" class=\"btn btn-light dropdown-toggle\" data-toggle=\"dropdown\" aria-haspopup=\"true\" aria-expanded=\"false\"> ".$_SESSION['user'][0]['pseudoUser']."</button>
                        <div class=\"dropdown-menu\" aria-labelledby=\"dropdownMenuButton\">
                            <a class=\"dropdown-item\" href=\"../controller/destroySession.php\">Déconnexion</a>
```

```
                        </div>
                    </div>";
    if($_SESSION['user'][0]['role_idrole'] == 0)
    {
        $navMenu1 = "<h5><a class=\"nav-link\" href=\"../view/homePage.php\">Accueil <span class=\"sr-only\">(current)</span></a></h5>";
        $navMenu2 = "<h5><a class=\"nav-link\" href=\"../view/tablatureManagerPage.php\">Gestion des tablatures </a></h5>";
    }
    elseif($_SESSION['user'][0]['role_idrole'] == 1)
    {
        $navMenu1 ="<h5><a class=\"nav-link\" href=\"../view/homePage.php\">Accueil <span class=\"sr-only\">(current)</span></a></h5>";
        $navMenu2 = "<h5><a class=\"nav-link\" href=\"../view/tablatureAndUserManagerPage.php\">Gestion des tablatures et utilisateurs </a></h5>";
    }
}
else
{
    $redirect = 0;
}
?>
<! DOCTYPE HTML>
<html>
<head>
    <title>Accueil</title>
    <script src="https://code.jquery.com/jquery-3.3.1.js" integrity="sha256-2Kok7MbOyxpgUVvAk/HJ2jigOSYS2auK4Pfzbm7uH60=" crossorigin="anonymous"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.3/umd/popper.min.js" integrity="sha384-ZMP7rVo3mIykV+2+9J3UJ46jBk0WLaUAdn689aCwoqbBJiSnjAK/l8WvCWPIPm49" crossorigin="anonymous"></script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.1/js/bootstrap.min.js" integrity="sha384-smHYKdLADwkXOn1EmN1qk/HfnUcbVRZyYmZ4qpPea6sjB/pTJ0euyQp0Mk8ck+5T" crossorigin="anonymous"></script>
    <link rel="stylesheet" type="text/css" href="../public/css/style.css">
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.0/css/bootstrap.min.css" integrity="sha384-9gVQ4dYFwwWSjIDZnLEWnxCjeSWFphJiwGPXr1jddIhOegiu1FwO5qRGvFXOdJZ4" crossorigin="anonymous">

    <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.0.13/css/all.css" integrity="sha384-DNOHZ68U8hZfKXOrtjWvjxusGo9WQnrNx2sqG0tfsghAvtVlRW3tvkXWZh58N9jp" crossorigin="anonymous">
    <!-- JQuery CDN -->
</head>
<body>
<div class="border ">
    <div class=".col text-right mr-3 mb-0"> <?php echo $navIdOrButton;?>

        <div id="body" class=" mx-5">
            <nav class="navbar navbar-expand-lg navbar-light p-0">
                <a class="navbar-brand" href="../view/homePage.php" id="logo">
                    <img src="../public/images/logoSimpleTabGrand.png" alt="logo" width="215" height="125">
                </a>

                <div class="collapse navbar-collapse" id="navbarSupportedContent">
                    <ul class="navbar-nav mr-auto " style="margin: auto;">
                        <li class="nav-item active">
                            <?php echo $navMenu1?>
                        </li>
                        <li class="nav-item active">
                            <?php echo $navMenu2?>
                        </li>
                    </ul>

                    <form class="form-inline my-2 my-lg-0" method="get" action="../view/search.php">

                        <input class="form-control mr-sm-0" type="search" placeholder="Rechercher" aria-label="Search" id="searchBar" name="nameArtistOrTitleTab">
                        <button type="submit" class=" btn btn-outline-secondary ">
                            <i class="fas fa-search"></i>
                        </button>

                    </form>
                </div>
            </nav>
        </div>
    </div>
</div>

</div>
<div class=" m-5">
    <table class="table table-dark">
        <thead style="color:red">
        <tr>
            <th colspan="4" class="text-center">Gestion des tablatures</th>
        </tr>
        <tr>
            <th scope="col">Artiste</th>
            <th scope="col">Titre</th>
            <th scope="col">Difficulté</th>
            <th scope="col">Note</th>
        </tr>
        </thead>
        <tbody id='tabs'>
```

```
106         </tbody>
107     </table>
108     <table class="table table-dark">
109         <thead style="color:red">
110         <tr>
111             <th colspan="4" class="text-center">Gestion des utilisateurs</th>
112         </tr>
113         <tr>
114             <th scope="col">Pseudo</th>
115             <th scope="col">Email</th>
116             <th scope="col">Nombre de tablatures ajoutées</th>
117             <th scope="col">Gestion</th>
118         </tr>
119         </thead>
120         <tbody id='users'>
121         </tbody>
122     </table>
123 </div>
124 </body>
125 <script src="../public/js/function.js"></script>
126 <script type="text/javascript">
127     $( document ).ready(function() {
128 
129         var redirect = <?php echo $redirect ?>
130 
131         if(redirect == 1) {
132 
133             getAllNonApprouvedTabs();
134 
135             getUsersAndNbTabPosted();
136 
137 
138 
139         }
140         else
141         {
142             window.location.href = "../view/homePage.php";
143         }
144 
145 
146     });
147 </script>
```